

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of :
Tomoo KIMURA et al. :
Serial No. NEW : **Attn: APPLICATION BRANCH**
Filed February 9, 2004 : **Attorney Docket No. 2004_0156A**
LOGIC CIRCUIT OPTIMIZING METHOD, :
LOGIC CIRCUIT OPTIMIZING DEVICE :
AND LOGIC CIRCUIT COMPOSING :
DEVICE :

CLAIM OF PRIORITY UNDER 35 USC 119

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

THE COMMISSIONER IS AUTHORIZED
TO CHARGE ANY DEFICIENCY IN THE
FEES FOR THIS PAPER TO DEPOSIT
ACCOUNT NO. 23-0975

Sir:

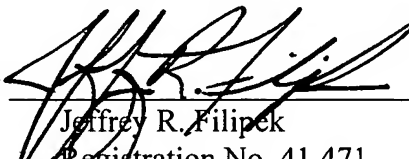
Applicants in the above-entitled application hereby claim the date of priority under the International Convention of Japanese Patent Application No. 2003-032438, filed February 10, 2003, as acknowledged in the Declaration of this application.

A certified copy of said Japanese Patent Application is submitted herewith.

Respectfully submitted,

Tomoo KIMURA et al.

By


Jeffrey R. Filipek
Registration No. 41,471
Attorney for Applicants

JRF/fs
Washington, D.C. 20006-1021
Telephone (202) 721-8200
Facsimile (202) 721-8250
February 9, 2004

日 本 国 特 許 庁

JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2003年 2月10日

出 願 番 号

Application Number:

特願2003-032438

[ST.10/C]:

[JP2003-032438]

出 願 人

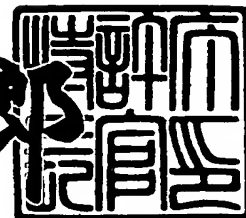
Applicant(s):

松下電器産業株式会社

2003年 6月19日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田信一郎



出証番号 出証特2003-3048151

【書類名】 特許願

【整理番号】 2038240106

【提出日】 平成15年 2月10日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/455
H03K 19/173 101

【発明者】

【住所又は居所】 大阪府門真市大字門真1006番地 松下電器産業株式
会社内

【氏名】 木村 智生

【発明者】

【住所又は居所】 大阪府門真市大字門真1006番地 松下電器産業株式
会社内

【氏名】 石田 健一

【発明者】

【住所又は居所】 大阪府門真市大字門真1006番地 松下電器産業株式
会社内

【氏名】 井本 智幸

【特許出願人】

【識別番号】 000005821

【氏名又は名称】 松下電器産業株式会社

【代理人】

【識別番号】 100097179

【弁理士】

【氏名又は名称】 平野 一幸

【手数料の表示】

【予納台帳番号】 058698

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0013529

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 論理回路最適化方法、論理回路最適化装置、及び、論理回路合成装置

【特許請求の範囲】

【請求項 1】 入力された論理回路情報に対してクラスタリングを行うクラスタリングステップと、

前記クラスタリングにより得られたクラスタのうち、クラスタ長が予め定められたクラスタ長を超えるクラスタに、フリップフロップを挿入する挿入ステップと、

前記フリップフロップを挿入した前記クラスタに対して、再クラスタリングを行う再クラスタリングステップと、を含む、論理回路最適化方法。

【請求項 2】 前記挿入ステップは、

前記クラスタリングにより得られた前記クラスタのクラスタ長を測定するクラスタ長測定ステップと、

前記クラスタリングにより得られた前記クラスタの中から、クラスタ長が前記予め定められたクラスタ長を超えるクラスタを選択するクラスタ選択ステップと

クラスタ長が前記予め定められたクラスタ長を超える前記クラスタに、フリップフロップを挿入するフリップフロップ挿入ステップと、を含み、

前記再クラスタリングにより得られたクラスタに、クラスタ長が前記予め定められたクラスタ長を超えるクラスタが存在する場合、前記フリップフロップ挿入ステップ、及び、前記再クラスタリングステップ、による処理を行う、請求項 1 記載の論理回路最適化方法。

【請求項 3】 前記クラスタリングおよび前記再クラスタリングを行った後の論理回路情報に含まれるクラスタを、論理エミュレーション装置の可変論理素子に割り付ける際に、前記可変論理素子に収まらないクラスタに、フリップフロップを挿入するステップと、

前記フリップフロップを挿入した前記クラスタに対して、再クラスタリングを行うステップと、をさらに含む、請求項 1 又は 2 記載の論理回路最適化方法。

【請求項 4】論理エミュレーション対象のメモリデバイスを実現するメモリデバイスを回路長で表すことにより、論理エミュレーション対象の前記メモリデバイスを論理回路に置き換えるステップ、をさらに含む、請求項 1 から 3 記載の論理回路最適化方法。

【請求項 5】前記回路長は、回路段数で表される、請求項 4 記載の論理回路最適化方法。

【請求項 6】前記回路長は、信号の伝搬時間で表される、請求項 4 記載の論理回路最適化方法。

【請求項 7】前記クラスタ長は、回路段数で表される、請求項 1 から 6 記載の論理回路最適化方法。

【請求項 8】前記クラスタ長は、信号の伝搬時間で表される、請求項 1 から 6 記載の論理回路最適化方法。

【請求項 9】クラスタリングを行った後の論理回路情報に含まれるクラスタを、論理エミュレーション装置の可変論理素子に割り付ける際に、前記可変論理素子に収まらないクラスタに、フリップフロップを挿入するステップと、

前記フリップフロップを挿入した前記クラスタに対して、再クラスタリングを行うステップと、を含む、論理回路最適化方法。

【請求項 10】挿入する前記フリップフロップの動作クロックの周波数は、前記フリップフロップの挿入の対象になった前記クラスタに含まれるフリップフロップの動作クロックの周波数より高い、請求項 1 から 9 記載の論理回路最適化方法。

【請求項 11】ハードウェア記述言語におけるモジュールに含まれる演算素子の数を算出するステップと、

予め定められた値より演算素子の数が多いモジュールに、フリップフロップを表すハードウェア記述言語を挿入するステップと、を含む、論理回路最適化方法。

【請求項 12】入力された論理回路情報に対してクラスタリングを行うクラスタリング手段と、

前記クラスタリングにより得られたクラスタのうち、クラスタ長が予め定めら

れたクラスタ長を超えるクラスタに、フリップフロップを挿入する回路分割手段と、を備え、

前記クラスタリング手段は、前記フリップフロップを挿入した前記クラスタに対して、再クラスタリングを行う、論理回路最適化装置。

【請求項 13】 前記クラスタリングにより得られた前記クラスタのクラスタ長を測定するクラスタ長測定手段と、

前記クラスタリングにより得られた前記クラスタの中から、前記予め定められたクラスタ長を超えるクラスタを選択するクラスタ選択手段と、をさらに備え、

前記再クラスタリングにより得られたクラスタに、前記予め定められたクラスタ長を超えるクラスタが存在する場合、前記回路分割手段は、当該クラスタに、フリップフロップを挿入する、請求項 12 記載の論理回路最適化装置。

【請求項 14】 前記回路分割手段は、前記クラスタリングおよび前記再クラスタリングを行った後の論理回路情報に含まれるクラスタを、論理エミュレーション装置の可変論理素子に割り付ける際に、前記可変論理素子に収まらないクラスタに、フリップフロップを挿入し、

前記クラスタリング手段は、前記フリップフロップを挿入した前記クラスタに対して、再クラスタリングを行う、請求項 12 又は 13 記載の論理回路最適化装置。

【請求項 15】 論理エミュレーション対象のメモリデバイスを実現するメモリデバイスを回路長で表すことにより、論理エミュレーション対象の前記メモリデバイスを論理回路に置き換える置換手段、をさらに備える、請求項 12 から 14 記載の論理回路最適化装置。

【請求項 16】 前記回路長は、回路段数で表される、請求項 15 記載の論理回路最適化装置。

【請求項 17】 前記回路長は、信号の伝搬時間で表される、請求項 15 記載の論理回路最適化装置。

【請求項 18】 前記クラスタ長は、回路段数で表される、請求項 12 から 17 記載の論理回路最適化装置。

【請求項 19】 前記クラスタ長は、信号の伝搬時間で表される、請求項 12 か

ら17記載の論理回路最適化装置。

【請求項20】 クラスタリングを行った後の論理回路情報に含まれるクラスタを、論理エミュレーション装置の可変論理素子に割り付ける際に、前記可変論理素子に収まらないクラスタに、フリップフロップを挿入する回路分割手段と、

前記フリップフロップを挿入した前記クラスタに対して、再クラスタリングを行うクラスタリング手段と、を備える、論理回路最適化装置。

【請求項21】 挿入する前記フリップフロップの動作クロックの周波数は、前記フリップフロップの挿入の対象になった前記クラスタに含まれるフリップフロップの動作クロックの周波数より高い、請求項12から20記載の論理回路最適化装置。

【請求項22】 ハードウェア記述言語におけるモジュールに含まれる演算素子の数を算出する演算素子数算出手段と、

予め定められた値より演算素子の数が多いモジュールに、フリップフロップを表すハードウェア記述言語を挿入するモジュール分割手段と、を備える、論理回路合成装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、半導体集積回路の論理検証を行う論理エミュレーション装置に与える回路情報を生成するための論理回路最適化方法およびその関連技術に関する。

【0002】

【従来の技術】

近年のLSI (large scale integrated circuit) の大規模化によって、ソフトウェア・シミュレーションによる論理回路の検証が不可能になってきている。

【0003】

そこで、ハードウェアによる回路動作を再現 (エミュレーション) する論理エミュレーション装置が実用化されている。

【0004】

論理エミュレーション装置には、大きく分けて2つのタイプがある。

1つは、多数のプロセッサを搭載して、強力な計算能力でエミュレーションするプロセッサ型論理エミュレーション装置である。

【0005】

もう1つは、FPGA型論理エミュレーション装置であり、ユーザが自在に回路動作を変えることができるFPGA (Field Programmable Gate Array) のようなプログラマブル・デバイス (可変論理素子) を多数搭載して、論理動作をエミュレーションする。

【0006】

従来のFPGA型エミュレーション装置の一例として、特許文献1に開示されているものがある。

【0007】

図19に示すように、この従来の論理エミュレーション装置100は、ロジックチップ101~103、メモリモジュール104、ユーザ指定チップ105、及び、相互接続チップ106、からなる。

【0008】

ロジックチップ101~103の各々がFPGAである。メモリモジュール104は、メモリデバイスをエミュレーションする。ユーザ指定チップ105は、ユーザが任意に使用するLSIなどのハードウェアIP (intellectual property) である。

【0009】

相互接続チップ106は、ロジックチップ101~103、メモリモジュール104、及び、ユーザ指定チップ105、を相互に接続する。この相互接続チップ106は、各チップ間をクロスバースイッチ構成で接続する。

【0010】

ここで、特許文献1に開示されている、FPGAに回路を割り付ける際に行われる基本的手法について説明する。

【0011】

論理回路は、基本的に、フリップフロップと、これらフリップフロップ間を接

続する組み合わせ回路と、からなっている。

【0012】

このため、FPGAへの論理回路の割り付けは、クラスタ単位で行われる。ここで、クラスタとは、フリップフロップの入力端子から前段のフリップフロップ（単数の場合もあるし、複数の場合もある。）の出力端子に向かって存在する組み合わせ回路と、当該フリップフロップと、から構成される回路群である。そして、クラスタを抽出する行為を、クラスタリングと呼ぶ。

【0013】

クラスタリングについて、図面を用いて説明する。

図20は、クラスタリングの説明図である。図20（a）は、クラスタリング前の論理回路の概念図、図20（b）は、クラスタリング後の論理回路の概念図、である。

【0014】

図20（a）では、フリップフロップFF0からフリップフロップFF1，FF2へ接続する回路構成例を示している。フリップフロップFF0とフリップフロップFF1，FF2とは、組み合わせ回路により接続される。

【0015】

この図20（a）の回路構成例において、クラスタリングを行うと、図20（b）に示すように、クラスタ107とクラスタ108とが作成される。

【0016】

クラスタ107は、組み合わせ回路109及びフリップフロップFF1からなる。クラスタ108は、組み合わせ回路110及びフリップフロップFF2からなる。

【0017】

このようなクラスタリングを、与えられた回路中に存在する全フリップフロップに対して実施する。このようにして作成されたクラスタをFPGAに割り付けていく。

【0018】

図21は、FPGAに割り付けられたクラスタの例示図である。図21に示す

ように、クラスタリング後に、クラスタ111がFPGA112に割り付けられる。

【0019】

さて、クラスタ単位でFPGAに論理回路を割り付けていく手法は、レンツ・ルールで示される、FPGAのI/O(input/output)数の不足問題を回避する目的がある。

【0020】

レンツ・ルールとは、1960年にIBMのE. F. Rent氏によって提唱された、論理設計におけるI/Oピン（ブロックのピンを含む）と回路規模との関係を示した法則である (http://www.cedcc.psu.edu/ee497i/rents_rule.PDF)。

【0021】

次式が、レンツ・ルールの関係式である。

【数1】

$$N_p = K_p \cdot N_g^x$$

（数1）において、「 N_p 」は、I/Oピン数、「 N_g 」は、回路規模（回路（ゲート）数）、を示す。「 x 」は、Rent定数、「 K_p 」は、比例定数、である。

【0022】

つまり、レンツ・ルールによれば、I/Oピン数は、回路規模の指数倍の関係で増える。

【0023】

【特許文献1】

特開2000-36737号公報（第1図）

【0024】

【発明が解決しようとする課題】

今日の半導体プロセスの微細化に伴い、LSIがますます大規模化しており、FPGAへの論理回路の割り付けが難しくなっている。

【0025】

一方、LSIの高速動作も求められており、フリップフロップ間の回路段数はそのまま、または削減される傾向にあるが、組み合わせ回路の並列度は広がっている。

【0026】

このため、クラスタリングを行っても、論理回路がFPGAに適切に割り付けられないことが発生している。

【0027】

例えば、FPGAに配置しようとするクラスタが、当該FPGAに配置可能な残りの回路規模より大きい場合、そのクラスタは、当該FPGAに配置されず、他のFPGAに配置されることになる。

【0028】

この場合、FPGAの使用効率が低下することになり、論理エミュレーション装置で取り扱える回路規模が小さくなるとう問題が生じる。

【0029】

また、FPGAの使用効率が低いと、割り付けられる回路密度が小さくなるため、各回路間の信号遅延が大きくなる。

【0030】

このため、信号遅延が付加され、動作速度つまりエミュレーション速度が低下するという問題が生じる。

【0031】

そこで、本発明は、論理エミュレーション装置の可変論理素子の使用効率を向上できる論理回路最適化方法及びその関連技術を提供することを目的とする。

【0032】

【課題を解決するための手段】

請求項1記載の論理回路最適化方法では、入力された論理回路情報に対してクラスタリングを行うクラスタリングステップと、クラスタリングにより得られたクラスタのうち、クラスタ長が予め定められたクラスタ長を超えるクラスタに、フリップフロップを挿入する挿入ステップと、フリップフロップを挿入したクラ

スタに対して、再クラスタリングを行う再クラスタリングステップと、を含む。

【0033】

この構成によれば、クラスタ長の長いクラスタにフリップフロップが挿入されるため、クラスタ長の長いクラスタが、クラスタ長の短い複数のクラスタに分割される。

【0034】

このため、論理エミュレーション装置の可変論理素子（例えば、FPGA）にクラスタを割り付ける際の自由度が高くなる。

【0035】

また、レンツ・ルールが示すような可変論理素子のI/Oピン数によるゲート搭載の非効率化が発生しないため、可変論理素子の使用効率を向上できる。

【0036】

以上により、論理エミュレーション装置に割り付ける論理回路の最適化が図られる。

【0037】

請求項2記載の論理回路最適化方法では、挿入ステップは、クラスタリングにより得られたクラスタのクラスタ長を測定するクラスタ長測定ステップと、クラスタリングにより得られたクラスタの中から、クラスタ長が予め定められたクラスタ長を超えるクラスタを選択するクラスタ選択ステップと、クラスタ長が予め定められたクラスタ長を超えるクラスタに、フリップフロップを挿入するフリップフロップ挿入ステップと、を含み、再クラスタリングにより得られたクラスタに、クラスタ長が予め定められたクラスタ長を超えるクラスタが存在する場合、フリップフロップ挿入ステップ、及び、再クラスタリングステップ、による処理を行う。

【0038】

この構成によれば、予め定められたクラスタ長を超えるクラスタについては、漏れなくフリップフロップが挿入されるため、最適化をより一層図ることができる。

【0039】

請求項3記載の論理回路最適化方法では、クラスタリングおよび再クラスタリングを行った後の論理回路情報に含まれるクラスタを、論理エミュレーション装置の可変論理素子に割り付ける際に、可変論理素子に収まらないクラスタに、フリップフロップを挿入するステップと、フリップフロップを挿入したクラスタに対して、再クラスタリングを行うステップと、をさらに含む。

【0040】

この構成によれば、論理エミュレーション装置に割り付ける論理回路のより一層の最適化を図ることができる。

【0041】

請求項4記載の論理回路最適化方法では、論理エミュレーション対象のメモリデバイスを実現するメモリデバイスを回路長で表すことにより、論理エミュレーション対象のメモリデバイスを論理回路に置き換えるステップ、をさらに含む。

【0042】

この構成によれば、論理エミュレーション対象のメモリデバイスについても、通常の回路と同様に、最適化処理を行うことができる。

【0043】

請求項5記載の論理回路最適化方法では、回路長は、回路段数で表される。

【0044】

この構成によれば、論理エミュレーション対象のメモリデバイスを実現するメモリデバイスが回路段数で表されるため、論理エミュレーション対象のメモリデバイスについても、通常の回路と同様に、最適化処理を行うことができる。

【0045】

請求項6記載の論理回路最適化方法では、回路長は、信号の伝搬時間で表される。

【0046】

この構成によれば、論理エミュレーション対象のメモリデバイスを実現するメモリデバイスが信号の伝搬時間で表されるため、論理エミュレーション対象のメモリデバイスについても、通常の回路と同様に、最適化処理を行うことができる。

【0047】

請求項7記載の論理回路最適化方法では、クラスタ長は、回路段数で表される。

【0048】

この構成によれば、回路段数の多いクラスタにフリップフロップが挿入されるため、回路段数の多いクラスタが、回路段数の少ない複数のクラスタに分割される。

【0049】

このため、可変論理素子にクラスタを割り付ける際の自由度が高くなり、可変論理素子の仕様効率の向上を図ることができる。

【0050】

請求項8記載の論理回路最適化方法では、クラスタ長は、信号の伝搬時間で表される。

【0051】

この構成によれば、信号の伝搬時間が長いクラスタにフリップフロップが挿入されるため、信号の伝搬時間が長いクラスタが、信号の伝搬時間が短い複数のクラスタに分割される。

【0052】

このため、可変論理素子にクラスタを割り付ける際の自由度が高くなり、可変論理素子の仕様効率の向上を図ることができる。

【0053】

請求項9記載の論理回路最適化方法では、クラスタリングを行った後の論理回路情報に含まれるクラスタを、論理エミュレーション装置の可変論理素子に割り付ける際に、可変論理素子に収まらないクラスタに、フリップフロップを挿入するステップと、フリップフロップを挿入したクラスタに対して、再クラスタリングを行うステップと、を含む。

【0054】

この構成によれば、可変論理素子（例えば、FPGA）に収まらないクラスタにフリップフロップが挿入されるため、可変論理素子に収まらないクラスタが、

クラスタ長の短い複数のクラスタに分割される。

【0055】

このため、クラスタ長の短いクラスタを、可変論理素子に割り付けることができ、可変論理素子の使用効率を向上できる。

【0056】

以上により、論理エミュレーション装置に割り付ける論理回路の最適化が図られる。

【0057】

請求項10記載の論理回路最適化方法では、挿入するフリップフロップの動作クロックの周波数は、フリップフロップの挿入の対象になったクラスタに含まれるフリップフロップの動作クロックの周波数より高い。

【0058】

この構成によれば、フリップフロップを挿入後でも、挿入前と同様の動作速度を維持できる。

【0059】

請求項11記載の論理回路最適化方法では、ハードウェア記述言語におけるモジュールに含まれる演算素子の数を算出するステップと、予め定められた値より演算素子の数が多いモジュールに、フリップフロップを表すハードウェア記述言語を挿入するステップと、を含む。

【0060】

この構成によれば、論理合成後のクラスタのクラスタ長が短くなるため、論理エミュレーション装置の可変論理素子（例えば、FPGA）にクラスタを割り付ける際の自由度が高くなる。

【0061】

また、レンツ・ルールが示すような可変論理素子のI/Oピン数によるゲート搭載の非効率化が発生しないため、可変論理素子の使用効率を向上できる。

【0062】

以上により、論理エミュレーション装置に割り付ける論理回路の最適化が図られる。

【0063】

さらに、ハードウェア記述言語段階でフリップフロップを挿入するため、論理合成後にフリップフロップを挿入する場合より、高速に処理することができる。

【0064】

【発明の実施の形態】

以下、図面を参照して本発明の実施の形態を説明する。

(実施の形態1)

【0065】

図1は、本発明の実施の形態1における論理エミュレーションシステムのブロック図である。

【0066】

図1に示すように、この論理エミュレーションシステムは、回路割付装置1、及び、論理エミュレーション装置2、を具備する。

【0067】

論理エミュレーション装置2には、図示していないが、FPGA及びメモリデバイス等が搭載されている。また、FPGAには、図示していないが、メモリデバイスが搭載されている。

【0068】

ここで、FPGAは、ユーザが自在に回路動作を変えることができる可変論理素子（プログラマブル・デバイス）の一例である。

【0069】

論理エミュレーション装置2として、例えば、図19の論理エミュレーション装置を使用できる。

【0070】

図2は、図1の回路割付装置1のブロック図である。なお、図2において、図1と同様の部分については、同一の符号を付している。

【0071】

図2に示すように、図1の回路割付装置1は、論理回路合成装置9、論理回路最適化装置8、及び、FPGA割付装置10、を含む。

【0072】

論理回路合成装置9は、論理合成を行い、ハードウェア記述言語（HDL：hardware description language）3から論理回路情報4を生成する。

【0073】

ハードウェア記述言語3は、機能レベルで論理回路を記述するための言語であり、例えば、Verilog-HDL、あるいはVHDLなど、である。

【0074】

論理回路情報4は、ゲート・レベルで論理回路を記述した情報であり、例えば、ネットリストである。

【0075】

図3は、図2の論理回路最適化装置8のブロック図である。図3に示すように、論理回路最適化装置8は、クラスタリング手段80、クラスタリング論理回路情報記憶手段81、クラスタ長測定手段82、クラスタ長記憶手段83、クラスタ選択手段84、回路分割手段85、回路分割論理回路情報記憶手段86、クラスタ長判定手段87、最適化論理回路情報生成手段88、及び、最適化論理回路情報記憶手段89、を含む。

【0076】

クラスタリング手段80は、論理回路合成装置9が生成した論理回路情報4の中からフリップフロップを検索し、全てのフリップフロップに対して、クラスタリング処理を行う。

【0077】

より具体的には、クラスタリング手段80は、現在選択しているフリップフロップの入力端子から前段のフリップフロップ（単数の場合もあるし、複数の場合もある。）の出力端子に向かって組み合わせ回路を探索して、現在選択しているフリップフロップ及び探索した組み合わせ回路で構成されるクラスタを抽出する。

【0078】

さて、クラスタリング論理回路情報記憶手段81は、クラスタリング後の論理

回路情報 4（以下、「クラスタリング論理回路情報」と呼ぶ。）を記憶する。

【0079】

クラスタ長測定手段 82 は、クラスタリング論理回路情報記憶手段 81 に記憶された全てのクラスタのクラスタ長を測定する。

【0080】

ここで、クラスタ長は、例えば、回路段数で表すことができる。
また、クラスタ長は、例えば、クラスタを信号が伝搬する際の伝搬時間で表すことができる。

【0081】

クラスタ長記憶手段 83 は、クラスタ長測定手段 82 が測定したクラスタ長を、各クラスタと関連づけて記憶する。

【0082】

クラスタ選択手段 84 は、クラスタ長記憶手段 83 を参照して、クラスタ長が規定値を超えているクラスタを選択する。

【0083】

ここで、上記の規定値は、全クラスタ長の平均値としたり、あるいは、論理エミュレーション装置 2 に最適な値としたり、などユーザが任意に指定できる。

【0084】

回路分割手段 85 は、クラスタ長が規定値を超えるクラスタの組み合わせ回路を分割するフリップフロップ（以下、「分割フリップフロップ」と呼ぶ。）を挿入する。

【0085】

なお、クラスタ長が規定値を超えるクラスタが複数存在する場合もあるが、この場合でも、クラスタ長が規定値を超える全てのクラスタに分割フリップフロップを挿入する。

【0086】

回路分割手段 85 による処理を具体例を挙げながら説明する。

図 4 は、回路分割手段 85 による処理の例示図である。図 4（a）は、回路分割前のクラスタを示し、図 4（b）は、回路分割後のクラスタを示している。

【0087】

図4 (a) に示すように、組み合わせ回路CC及びフリップフロップFF1からなるクラスタCのクラスタ長は、規定値を超えているとする。

【0088】

回路分割手段85が、クラスタCに分割フリップフロップFF2を挿入し、クラスタリング手段80が、分割フリップフロップFF2を挿入したクラスタCに対して、再クラスタリングを行う。

【0089】

その結果、図4 (b) に示すように、クラスタCは、組み合わせ回路CC1及びフリップフロップFF2からなるクラスタC1と、組み合わせ回路CC2及びフリップフロップFF1からなるクラスタC2と、に分割される。

【0090】

ここで、分割フリップフロップFF2は、他のフリップフロップFF0, FF1の動作クロックCLKとは異なる動作クロックFCLKで動作する。この点を詳しく説明する。

【0091】

図4 (a) に示すように、フリップフロップFF0-FF1間の組み合わせ回路CCは、動作クロックCLKの1クロックで動作する。

【0092】

従って、図4 (b) に示すように、分割フリップフロップFF2を挿入後でも、フリップフロップFF0-FF1間では、動作クロックCLKの1クロックで動作しなければならない。

【0093】

そこで、分割フリップフロップFF2の動作クロックFCLKを、動作クロックCLKの立ち下がリエッジに同期したクロックとする。なお、フリップフロップFF0, FF1および分割フリップフロップFF2は、立ち上がりエッジ動作を行うとする。

【0094】

例えば、動作クロックFCLKを、動作クロックCLKの反転クロックとする

ことができる。

【0095】

また、例えば、動作クロック FCLK を、動作クロック CLK の立ち下がリエッジに同期したパルス波形のクロックとすることができる。

【0096】

このように、動作クロック FCLK を動作クロック CLK の立ち下がリエッジに同期したクロックとすると、フリップフロップ FF0-分割フリップフロップ FF2 間の組み合わせ回路 CC1、及び、分割フリップフロップ FF2-フリップフロップ FF1 間の組み合わせ回路 CC2、の各々は、動作クロック CLK の半周期以内で動作する必要がある。

【0097】

従って、組み合わせ回路 CC1 及び組み合わせ回路 CC2 の各々が、動作クロック CLK の半周期以内で動作する回路段数のポイントに、分割フリップフロップ FF2 を挿入する必要がある。

【0098】

さて、分割フリップフロップ FF2 に対しては、フリップフロップ機能の許可（イネーブル）制御が可能である。

【0099】

つまり、分割フリップフロップ FF2 は、フリップフロップ機能のイネーブル信号 ENB によって、ラッチ動作又はスルー動作を切り替えることができる。従って、ディスイネーブル時は、分割フリップフロップ FF2 の挿入前の回路（図 4（a））と同等の回路構成に戻して動作させることができる。

【0100】

このため、分割フリップフロップ FF2 を挿入する場合と挿入しない場合との論理等価性を確認することができる。

【0101】

さて、上記のように、クラスタリング手段 80 は、分割フリップフロップ FF2 を挿入したクラスタ C に対して、再クラスタリングを行って、クラスタ C1、C2 を作成した。

【0102】

この際、クラスタリング手段80は、クラスタリング論理回路情報記憶手段81に記憶されたクラスタリング論理回路情報において、クラスタCの情報をクラスタC1、C2の情報に置き換えて、クラスタリング論理回路情報を更新する。

【0103】

さて、図3のクラスタ長判定手段87は、フリップフロップFF0-分割フリップフロップFF2間、及び、分割フリップフロップFF2-フリップフロップFF1間、の各々のクラスタ長と、規定値と、を比較する。

【0104】

そして、クラスタ長判定手段87は、フリップフロップFF0-分割フリップフロップFF2間、あるいは、分割フリップフロップFF2-フリップフロップFF1間、のいずれか一方のクラスタ長が、規定値を超えている場合は、回路分割手段85、及び、クラスタリング手段80、に対して、それぞれ、回路分割処理、及び、再クラスタリング処理、を実行させる。

【0105】

回路分割処理、及び、再クラスタリング処理は、フリップフロップFF0-分割フリップフロップFF2間、及び、分割フリップフロップFF2-フリップフロップFF1間、の各々のクラスタ長が、規定値以下になるまで繰り返される。こうすることで、全てのクラスタのクラスタ長が規定値以下となる。

【0106】

ここで、クラスタ長判定手段87における規定値と、クラスタ選択手段84における規定値と、は同一の値とする。

【0107】

一方、フリップフロップFF0-分割フリップフロップFF2間、及び、分割フリップフロップFF2-フリップフロップFF1間、の各々のクラスタ長が、規定値以下の場合は、クラスタ長判定手段87は、図3の最適化論理回路情報生成手段88に、処理を実行させる。

【0108】

さて、再び、図3に戻って説明する。

回路分割論理回路情報記憶手段 8 6 は、最新の更新されたクラスタリング回路情報（以下、「回路分割論理回路情報」と呼ぶ。）を記憶する。

【 0 1 0 9 】

最適化論理回路情報生成手段 8 8 は、回路分割論理回路情報および論理エミュレーション装置構成情報 6 を基に、論理エミュレーション装置 2 に搭載された各 F P G A に、回路分割論理回路情報に含まれるクラスタを割り付けていく。

【 0 1 1 0 】

そして、最適化論理回路情報生成手段 8 8 は、クラスタの割り付け情報と回路分割論理回路情報とからなる最適化論理回路情報 1 1 を生成する。

【 0 1 1 1 】

ここで、論理エミュレーション装置構成情報 6 は、論理エミュレーション装置 2 に搭載されている素子に関する情報である。

【 0 1 1 2 】

例えば、論理エミュレーション装置構成情報 6 は、論理エミュレーション装置 2 に搭載されている F P G A の個数、種類（例えば、搭載可能な回路（ゲート）数（搭載可能な回路規模）など）、及び、接続関係、を示した情報である。

【 0 1 1 3 】

最適化論理回路情報記憶手段 8 9 は、最適化論理回路情報 1 1 を記憶する。
さて、再び、図 2 に戻って説明する。

【 0 1 1 4 】

F P G A 割付装置 1 0 は、最適化論理回路情報 1 1 を元に、論理エミュレーション装置用回路情報 5 を生成する。

【 0 1 1 5 】

論理エミュレーション装置用回路情報 5 は、最適化論理回路情報 1 1 が最終的な回路割付状態になった回路情報である。

【 0 1 1 6 】

図 1 の論理エミュレーション装置 2 は、この論理エミュレーション装置用回路情報 5 を基に、各 F P G A にクラスタを配置して、エミュレーションを実行する

【0117】

さて、次に、分割フリップフロップの挿入方法を、具体例を挙げながら詳細に説明する。この例では、クラスタ長を回路段数で表す。

【0118】

図5は、分割フリップフロップの挿入方法を説明するためのクラスタの例示図である。なお、図5のフリップフロップFF0～FF4は、立ち上がりエッジ動作を行うとする。

【0119】

図5には、フリップフロップFF4と、フリップフロップFF4-FF0, FF1, FF2, FF3間の組み合わせ回路と、からなる1つのクラスタを示している。

【0120】

このように、基点となるフリップフロップFF4に対して、その前段のフリップフロップFF0～FF3が複数存在する場合もある。

【0121】

なお、図4では、基点となるフリップフロップFF1に対して、その前段のフリップフロップFF0が1つ存在する例を挙げている。

【0122】

図6は、図5のクラスタのタイミング図である。図6には、フリップフロップFF0～FF4の動作クロックCLK, 分割フリップフロップの動作クロックFCLK, フリップフロップFF0～FF3への入力信号IN0～IN3, 図5のクラスタの各ノードN2, N4, N6, N6#, N7での信号波形、及び、フリップフロップFF4の出力信号OUT、が示されている。

【0123】

図6では、図5のクラスタに挿入する分割フリップフロップの動作クロックFCLKが、フリップフロップFF0～FF4の動作クロックCLKの立ち下がりエッジに同期したパルス波形のクロックである例を挙げている。

【0124】

従って、分割フリップフロップを挿入した場合に、その前後の組み合わせ回路の各々が、動作クロックCLKの半周期で動作するような位置に、分割フリップフロップを挿入する必要がある。

【0125】

そこで、図5のクラスタの最大の回路段数が4段であるため、その中間位置である2段目の位置に、分割フリップフロップを挿入する。

【0126】

ここで、フリップフロップFF0～FF3からフリップフロップFF4に向かって、素子を1つ越える度に、回路段数が1段上がっていく。

【0127】

さて、2段目の位置に分割フリップフロップを挿入したが、3段目の位置に分割フリップフロップを挿入しても、動作速度上問題がない場合は、3段目の位置に分割フリップフロップを挿入する。

【0128】

図5の入力信号IN2から出力信号OUTまでの信号遷移に注目して、動作速度を検討してみる。

【0129】

図6に示すように、入力信号IN0, IN1, IN2, IN3が、それぞれ、ハイ(high)固定、ロー(low)固定、ローからハイへの遷移、ロー固定、とする。

【0130】

ノードN2は、フリップフロップFF2の出力だから、動作クロックCLKの立ち上がりエッジで入力信号IN2をラッチする。

【0131】

そして、インバータ素子INV1、NAND素子NAND0、及び、AND素子AND1、を経由して、信号が伝搬する。

【0132】

ノードN6での信号遷移が、動作クロックCLKの半周期以内で終わっているため、動作クロックCLKの立ち上がりエッジで動作する3段目の位置に挿入

した分割フリップフロップでノードN 6の信号をラッチしても信号遅延による問題は生じない。

【0 1 3 3】

よって、ノードN 6 #, N 7では、図6のような波形を得ることができ、動作クロックCLKの1周期に対する回路動作の透過性を保ったままでクラスタを分割できる。

【0 1 3 4】

以上のことをまとめる。ノードN 2, N 4, N 6では、時刻t 1の動作クロックCLKの立ち上がりエッジに応答して、時刻t 2より以前に信号波形が遷移している。

【0 1 3 5】

つまり、分割フリップフロップの前段の組み合わせ回路は、動作クロックCLKの半周期以内で動作している。

【0 1 3 6】

一方、ノードN 6 #, N 7では、時刻t 2の動作クロックCLKの立ち上がりエッジに応答して、時刻t 3より以前に信号波形が遷移している。

【0 1 3 7】

つまり、分割フリップフロップの後段の組み合わせ回路は、動作クロックCLKの半周期以内で動作している。

【0 1 3 8】

よって、3段目の位置に分割フリップフロップを挿入しても、動作速度上問題がないことが分かる。

【0 1 3 9】

なお、時刻t 3の動作クロックCLKの立ち上がりエッジに応答して、出力信号OUTが出力される。

【0 1 4 0】

ここで、セクタMUX 0の出力選択信号は、前段のフリップフロップFF 3から直接与えられており、途中に素子が存在しない。

【0 1 4 1】

従って、セクタMUX0-フリップフロップFF3間には、分割フリップフロップを挿入する必要がない。

【0142】

よって、最終的には、セクタMUX0-EXOR素子XOR0間、及び、セクタMUX0-AND素子AND1間、に分割フリップフロップを挿入する。

【0143】

クラスタの2段目に分割フリップフロップを挿入する場合は、計3個の分割フリップフロップが必要であるが、クラスタの3段目に分割フリップフロップを挿入する場合は、計2個の分割フリップフロップを挿入するだけでよい。

【0144】

さて、次に、図3及びフローチャートを用いて、論理回路最適化装置8による処理の流れを説明する。

【0145】

図7は、図3の論理回路最適化装置8のフローチャートである。

図7に示すように、ステップS1にて、クラスタリング手段80は、エミュレーション対象の論理回路情報4に対して、クラスタリングを実行する。

【0146】

ステップS2にて、クラスタ長測定手段82は、クラスタリングで得られた全クラスタのクラスタ長を測定する。

【0147】

ステップS3にて、クラスタ選択手段84は、クラスタ長が規定値を超えるクラスタを選択する。

【0148】

ステップS4にて、回路分割手段85は、クラスタ長が規定値を超えるクラスタに、分割フリップフロップを挿入して、回路分割を行う。

【0149】

ステップS5にて、クラスタリング手段80は、分割フリップフロップが挿入されたクラスタに対して、再クラスタリングを行う。

【0150】

ステップS6にて、クラスタリング手段80は、再クラスタリングの結果に基づき、クラスタリング論理回路情報を更新する。

【0151】

ステップS7にて、クラスタ長判定手段87は、再クラスタリングにより得られたクラスタのクラスタ長が規定値以下かどうかを判定する。

【0152】

そして、再クラスタリングにより得られたクラスタのクラスタ長が規定値以下の場合は、ステップS8へ進む（ステップS7）。

【0153】

ステップS8では、最適化論理回路情報生成手段88は、回路分割論理回路情報および論理エミュレーション装置構成情報6を基に、最適化論理回路情報11を生成する。

【0154】

一方、再クラスタリングにより得られたクラスタのクラスタ長が規定値を超えている場合は、ステップS4へ進み（ステップS7）、再びステップS4からステップS7の処理を行い、クラスタリング論理回路情報を更新する。そして、最後に更新されたクラスタリング論理回路情報が、回路分割論理回路情報である。

【0155】

さて、以上のように、本実施の形態では、クラスタ長の長いクラスタに分割フリップフロップが挿入されるため、クラスタ長の長いクラスタが、クラスタ長の短い複数のクラスタに分割される。

【0156】

このため、論理エミュレーション装置のFPGAにクラスタを割り付ける際の自由度が高くなって、FPGAの使用効率を向上できる。

【0157】

これにより、論理エミュレーション装置に割り付ける論理回路の最適化が図られる。

【0158】

また、FPGAの使用効率の向上により、回路密集度が高くなって、各回路素

子の信号遅延が少なくなる。その結果、エミュレーション速度が向上する。

【0159】

なお、図1の回路割付装置1は、コンピュータシステムで稼働するプログラムで実現することができる。ただし、プログラムで実現する場合に限られない。

【0160】

また、FPGA割付装置10は、FPGAコンパイラと呼ばれるプログラムで実現できる。ただし、プログラムで実現する場合に限られない。

【0161】

(実施の形態2)

本発明の実施の形態2における論理エミュレーションシステムの全体構成は、図1と同様である。従って、図1の論理エミュレーションシステムを、実施の形態2による論理エミュレーションシステムとして説明する。

【0162】

また、実施の形態2による回路割付装置1の構成は、図2と同様である。従って、図2の回路割付装置1を、実施の形態2による回路割付装置1として説明する。

【0163】

また、実施の形態2による論理回路最適化装置8は、図3と同様である。従って、図3の論理回路最適化装置8を、実施の形態2による論理回路最適化装置8として説明する。

【0164】

図8は、実施の形態2による論理回路最適化装置8の説明図である。図8(a)は、FPGA20にクラスタ22が収まらない状態を示し、図8(b)は、FPGA20にクラスタ群23が収まった状態を示している。

【0165】

図3の最適化論理回路情報生成手段88は、論理エミュレーション装置構成情報6を基に、回路分割論理回路情報に含まれる各クラスタを、最終的にどのFPGAに割り付けるかを決定して、最適化論理回路情報11を生成する。

【0166】

しかしながら、図 8 (a) に示すように、FPGA 20 の境界部分にクラスタ 22 が割り付けられたりする等、FPGA 20 に割り付けようとするクラスタ 22 が、FPGA 20 に配置可能な残りの回路規模より大きい場合もある。

【0167】

この場合に、実施の形態 2 では、図 8 (a) のクラスタ 22 に分割フリップフロップを挿入して、クラスタ 22 を細分化した複数のクラスタを生成し、図 8 (b) に示すように、FPGA 20 に配置可能な数のクラスタからなるクラスタ群 23 を、FPGA 20 に割り付ける。

【0168】

具体的には、FPGA 20 に割り付けようとするクラスタ 22 が、FPGA 20 に配置可能な残りの回路規模より大きい場合は、回路分割手段 85 が、クラスタ 22 に、分割フリップフロップを挿入する。

【0169】

そして、クラスタリング手段 80 は、分割フリップフロップを挿入したクラスタ 22 に対して、再クラスタリングを行い、回路分割論理回路情報を更新する。

【0170】

そして、最適化論理回路情報生成手段 88 は、FPGA 20 の空き領域に配置可能な回路規模（回路（ゲート）数）および FPGA 20 の I/O 数を決定条件として、クラスタ 22 を分割して得られた複数のクラスタに対して、再び割り付け処理を実行し、最適化論理回路情報 11 を更新する。

【0171】

その結果、FPGA 20 に配置可能な残りの回路規模に収まるクラスタ群 23 が得られ、FPGA 20 に配置される。

【0172】

実施の形態 2 の以上の点以外については、実施の形態 1 と同様であり、説明を省略する。

【0173】

従来は、FPGA に配置しようとするクラスタが、当該 FPGA に配置可能な残りの回路規模より大きい場合は、他の FPGA に配置されていたが、本実施の

形態では、クラスタが F P G A の空き領域に収まらない場合は、上記のように、そのクラスタを細分化することにより、F P G A の使用効率を向上できる。

【 0 1 7 4 】

さて、本実施の形態による論理回路最適化装置 8 による処理の流れをフローチャートを用いて説明する。

【 0 1 7 5 】

本実施の形態による論理回路最適化装置 8 による処理の流れは、図 7 に示したフローチャートと同様である。

【 0 1 7 6 】

ただし、本実施の形態では、図 7 のフローチャートのステップ S 8 での最適化論理回路情報生成処理が異なっており、図 9 のフローチャートに示す最適化論理回路情報生成処理が行われる。

【 0 1 7 7 】

図 9 は、本実施の形態の論理回路最適化装置 8 による最適化論理回路情報生成処理のフローチャートである。

【 0 1 7 8 】

前提として、図 7 に示したステップ S 1 からステップ S 7 までの処理が終了しているとする。

【 0 1 7 9 】

さて、図 9 に示すように、ステップ S 8 1 にて、最適化論理回路情報生成手段 8 8 は、回路分割論理回路情報に含まれる 1 つのクラスタを選択する。

【 0 1 8 0 】

ステップ S 8 2 にて、最適化論理回路情報生成手段 8 8 は、選択したクラスタを F P G A に割り付ける。

【 0 1 8 1 】

ステップ S 8 3 にて、最適化論理回路情報生成手段 8 8 は、割り付けたクラスタが、F P G A に収まっているかどうかを判定する。

【 0 1 8 2 】

割り付けたクラスタが、F P G A に収まっている場合は（ステップ S 8 3）、

ステップ S 8 4 にて、最適化論理回路情報生成手段 8 8 は、そのクラスタの割り付け情報を確定する。

【0183】

一方、割り付けたクラスタが、FPGA に収まらない場合は（ステップ S 8 3）、ステップ S 8 6 に進む。

【0184】

ステップ S 8 6 にて、最適化論理回路情報生成手段 8 8 は、収まらないクラスタに、分割フリップフロップが挿入されているかどうかを判断する。

【0185】

収まらないクラスタに分割フリップフロップが挿入されている場合は（ステップ S 8 6）、最適化論理回路情報生成手段 8 8 は、ステップ S 8 7 に進む。

【0186】

ステップ S 8 7 にて、最適化論理回路情報生成手段 8 8 は、その収まらないクラスタを別の FPGA に割り付け、ステップ S 8 4 に進み、そのクラスタの割り付け情報を確定する。

【0187】

一方、収まらないクラスタに分割フリップフロップが挿入されていない場合は（ステップ S 8 6）、最適化論理回路情報生成手段 8 8 は、ステップ S 8 8 に進む。

【0188】

ステップ S 8 8 にて、最適化論理回路情報生成手段 8 8 は、回路分割手段 8 5 に対して、その収まらないクラスタに、分割フリップフロップを挿入するように指示する。

【0189】

この指示を受けて、回路分割手段 8 5 は、その収まらないクラスタに分割フリップフロップを挿入する。

【0190】

この場合の分割フリップフロップの挿入方法は、図 7 のステップ S 4 の回路分割処理と同様であり、動作速度上問題のない位置に分割フリップフロップを挿入

して、クラスタを細分化する。細分化の数（分割数）は、任意に設定できる。

【0191】

ステップS89にて、クラスタリング手段80は、分割フリップフロップが挿入されたクラスタに対して、再クラスタリングを行い、回路分割論理回路情報を更新する。

【0192】

ステップS90にて、最適化論理回路情報生成手段88は、細分化して得られたクラスタをFPGAの空き領域に割り付け、ステップS84に進み、細分化して得られたクラスタの割り付け情報を確定する。

【0193】

さて、ステップS85にて、次のクラスタが存在する場合は、ステップS81に進み、上記の処理が実行される。

【0194】

このように、順次クラスタを更新しながら、上記の処理を実行して、全てのクラスタの割り付け情報を確定していき、最新の回路分割論理回路情報および確定した割り付け情報からなる最適化論理回路情報11を生成する。

【0195】

なお、図9では、分割フリップフロップの動作クロックFCLKが、通常のフリップフロップの動作クロックCLKの立ち下がリエッジに同期したパルス波形のクロックである例を挙げている。

【0196】

従って、FPGAに収まらないクラスタに既に分割フリップフロップが挿入されている場合は（ステップS86）、このクラスタに対しては、さらに、分割フリップフロップを挿入することができないため、ステップS87の処理が実行される。

【0197】

さて、以上のように、本実施の形態では、実施の形態1と同様の構成を具備しているため、実施の形態1と同様に、クラスタ長の長いクラスタが、クラスタ長の短い複数のクラスタに分割される。

【 0 1 9 8 】

加えて、本実施の形態では、FPGAに収まらないクラスタにフリップフロップが挿入されるため、FPGAに収まらないクラスタが、クラスタ長の短い複数のクラスタに分割される。

よって、FPGAの使用効率をより一層向上できる。

【 0 1 9 9 】

(実施の形態 3)

実施の形態 1、及び、実施の形態 2 では、論理回路合成装置 9 が生成した論理回路情報 4 に対して、最適化処理を実行した。

【 0 2 0 0 】

実施の形態 3 では、論理合成前、即ち、ハードウェア記述言語に対して、最適化処理を実行する。

【 0 2 0 1 】

さて、本発明の実施の形態 3 における論理エミュレーションシステムの全体構成は、図 1 と同様である。従って、図 1 の論理エミュレーションシステムを、実施の形態 3 による論理エミュレーションシステムとして説明する。

【 0 2 0 2 】

また、実施の形態 3 による回路割付装置 1 の構成は、図 2 と同様である。従って、図 2 の回路割付装置 1 を、実施の形態 3 による回路割付装置 1 として説明する。

【 0 2 0 3 】

ただし、実施の形態 3 では、論理回路合成装置 9 および論理回路最適化装置 8 の構成は、実施の形態 1 と異なる。その他の点は、実施の形態 1 と同様である。

【 0 2 0 4 】

図 1 0 は、本発明の実施の形態 3 における論理回路合成装置 9 のブロック図である。

【 0 2 0 5 】

図 1 0 に示すように、この論理回路合成装置 9 は、演算素子数算出手段 9 3、モジュール分割手段 9 0、HDL 記憶手段 9 1、及び、合成手段 9 2、を含む。

【0206】

演算素子数算出手段93は、ハードウェア記述言語を取り込んで、各モジュールの機能集合（演算素子の塊）を解析して、モジュール毎に入力端子から出力端子までの演算素子の数を算出する。

【0207】

この場合の、入力端子、出力端子、及び演算素子は、ハードウェア記述言語で表されている。

【0208】

そして、モジュール分割手段90は、処理が長い経路、即ち、入力端子から出力端子までの演算素子の数が多い経路、を有するモジュール選択する。

【0209】

この場合、入力端子から出力端子までの演算素子の数が多いかどうかは、規定値と比較して判定される。この規定値は、ユーザが任意に設定できる。

【0210】

そして、モジュール分割手段90は、入力端子から出力端子までの演算素子の数が多い経路を有するモジュールに、分割フリップフロップを表すハードウェア記述言語を挿入する。

【0211】

この場合の分割フリップフロップは、実施の形態1の分割フリップフロップと同様のものである。

【0212】

このように、分割フリップフロップを表すハードウェア記述言語を「機能分割フリップフロップ」と呼ぶ。

【0213】

以上のようにして、モジュール分割手段90は、最初に入力されたハードウェア記述言語に対して、機能分割フリップフロップを挿入していき、ハードウェア記述言語を更新する。

【0214】

HDL記憶手段91は、更新後のハードウェア記述言語を記憶する。

合成手段92は、論理合成を行って、更新後のハードウェア記述言語から論理回路情報4を生成する。

【0215】

実施の形態3の論理回路最適化装置8は、論理回路情報4に対して、クラスタリングを行い、クラスタリング論理回路情報を生成する。

【0216】

そして、論理回路最適化装置8は、クラスタリング論理回路情報に含まれる各クラスタを、論理エミュレーション装置構成情報6を参照しながら、FPGAに割り付けていく。

【0217】

このようにして、論理回路最適化装置8は、クラスタリング論理回路情報及びクラスタの割り付け情報からなる最適化論理回路情報11を生成する。その後の処理は、実施の形態1と同様である。

【0218】

さて、図10の論理回路合成装置9の動作を具体例を挙げながら説明する。

図11は、機能分割フリップフロップ挿入前のハードウェア記述言語におけるモジュールの例示図である。

図12は、図11のモジュールのツリー図である。

図13は、機能分割フリップフロップ挿入後のハードウェア記述言語におけるモジュールの例示図である。

【0219】

演算素子数算出手段93に入力されたハードウェア記述言語が、図11に示すモジュールを含むとする。

【0220】

このモジュールは、機能集合40、41を含む。機能集合40から明らかなように、入力Aは、3つの演算処理をまたがって、出力OUTに接続される。

【0221】

演算素子数算出手段93は、このような解析を、他の入力B、Cに対しても実行する。

【 0 2 2 2 】

そうすると、図 1 2 に示すツリー図が得られる。なお、このツリー図は説明の便宜のためのものであり、実際に作成されるわけではない。

【 0 2 2 3 】

モジュール分割手段 9 0 は、演算素子数算出手段 9 3 による解析の結果得られた入力 A, B, C から出力 O U T までの演算素子の数と、規定値と、を比較して、入力 A, B, C から出力 O U T までの演算素子の数が多いと判断した場合は、図 1 3 に示すように、このモジュールに、機能分割フリップフロップ 4 2 を挿入する。

【 0 2 2 4 】

なお、図 1 2 のツリー図では、矢印 \$ で示した位置に機能分割フリップフロップが挿入される。

【 0 2 2 5 】

さて、次に、図 1 0 の論理回路合成装置 9 による処理の流れをフローチャートを用いて説明する。

【 0 2 2 6 】

図 1 4 は、図 1 0 の論理回路合成装置 9 のフローチャートである。
図 1 4 に示すように、ステップ S 2 1 にて、演算素子数算出手段 9 3 は、ハードウェア記述言語に含まれる 1 つのモジュールを選択する。

【 0 2 2 7 】

ステップ S 2 2 にて、演算素子数算出手段 9 3 は、入力から出力までの演算素子数を算出する。

【 0 2 2 8 】

ステップ S 2 3 にて、モジュール分割手段 9 0 は、演算素子数算出手段 9 3 が算出した演算素子数を規定値と比較して、演算素子数が規定値以下であれば、ステップ S 2 5 に進む。

【 0 2 2 9 】

一方、ステップ S 2 3 にて、モジュール分割手段 9 0 は、演算素子数算出手段 9 3 が算出した演算素子数を規定値と比較して、演算素子数が規定値を超えてい

れば、ステップ S 24 に進む。

【0230】

ステップ S 24 にて、モジュール分割手段 90 は、演算素子数が規定値を超えるモジュールに、機能分割フリップフロップを挿入する。

【0231】

ステップ S 25 にて、次のモジュールがある場合は、ステップ S 21 に進み、ステップ S 21 からステップ S 25 の処理を行う。

【0232】

以上のような処理を繰り返して、機能分割フリップフロップを挿入していき、ハードウェア記述言語を更新する。

【0233】

一方、ステップ S 25 にて、次のモジュールがない場合は、ステップ S 26 に進み、合成手段 92 が、更新後のハードウェア記述言語から論理回路情報 4 を生成する。

【0234】

さて、以上のように、本実施の形態では、規定値より演算素子の数が多いモジュールに、分割フリップフロップを表すハードウェア記述言語を挿入する。

【0235】

このため、論理合成後のクラスタのクラスタ長が短くなり、論理エミュレーション装置の F P G A にクラスタを割り付ける際の自由度が高くなる。よって、可変論理素子の使用効率を向上できる。

【0236】

これにより、論理エミュレーション装置に割り付ける論理回路の最適化が図られる。

【0237】

さらに、ハードウェア記述言語段階で分割フリップフロップを挿入するため、論理合成後に分割フリップフロップを挿入する場合より、高速に処理することができる。

【0238】

(実施の形態4)

本発明の実施の形態4における論理エミュレーションシステムの全体構成は、図1と同様である。従って、図1の論理エミュレーションシステムを、実施の形態4による論理エミュレーションシステムとして説明する。

【0239】

また、実施の形態4による回路割付装置1の構成は、図2と同様である。従って、図2の回路割付装置1を、実施の形態4による回路割付装置1として説明する。ただし、実施の形態4の論理回路最適化装置8の構成は、実施の形態1の論理回路最適化装置8の構成と異なる。

【0240】

図15は、本発明の実施の形態4における論理回路最適化装置8のブロック図である。なお、図15において、図3と同様の部分については、同一の符号を付して、説明を適宜省略する。

【0241】

図15に示すように、この論理回路最適化装置8は、図3の論理回路最適化装置8の構成に、置換手段95、を追加したものである。

【0242】

さて、設計する回路には、様々なメモリデバイスが含まれる。例えば、SRAM (static random access memory) やSDRAM (synchronous dynamic random access memory) などがあり、当然ながらそれぞれの動作仕様は異なる。

【0243】

そして、論理エミュレーション装置2及びFPGAは、SRAM等の固定のメモリデバイスを搭載している。

【0244】

そこで、実現したいメモリデバイスの動作仕様に合わせるため、論理エミュレーション装置2あるいはFPGAに搭載された固定のメモリデバイスに、周辺のモデル回路（ラッパ回路）を付加して、メモリデバイスを実現する。

【0245】

そして、置換手段95は、メモリ置換テーブル情報を参照して、論理エミュレーション装置2あるいはFPGAに搭載された固定のメモリデバイスを回路長（例えば、回路段数あるいは信号の伝搬時間）で表すことにより、上記のようにして実現したメモリデバイスを論理回路に置き換える。

【0246】

以上の点を具体例を挙げながら説明する。

図16は、図15の置換手段95の動作の説明図である。

図17は、メモリ置換テーブル情報の例示図である。

【0247】

図16に示すように、実現したいメモリデバイスであるSDRAM34の動作仕様に合わせるため、論理エミュレーション装置2あるいはFPGAに搭載された固定のメモリデバイスであるSRAM36に、組み合わせ回路31、32からなるラッパー回路35を付加して、SDRAM34を実現する。なお、SDRAM34は、組み合わせ回路30、33に接続される。

【0248】

このようにして実現されたSDRAM34の情報が、論理回路情報4に含まれている。

【0249】

そして、置換手段95は、図17に示すようなメモリ置換テーブル情報を参照して、論理回路情報4に含まれるSRAM36を回路長で表すことにより、SDRAM34を論理回路に置き換える。

【0250】

例えば、SRAM36が、4MbitのSRAMならば、図17のメモリ置換テーブル情報によれば、このSRAM36は、回路段数が「10」の論理回路とみなせる。

【0251】

そうすると、SRAM36にラッパー回路35を付加して作成したSDRAM34は、通常の組み合わせ回路とみなすことができる。

【0252】

従って、その後は、図 1 5 の論理回路最適化装置 8 により、実施の形態 1 と同様の処理を行って、最適化を実行する。

【 0 2 5 3 】

なお、図 1 7 のメモリ置換テーブル情報では、換算値として、回路段数を採用しているが、信号の伝搬時間を採用することもできる。

【 0 2 5 4 】

さて、次に、図 1 5 の論理回路最適化装置 8 による処理の流れをフローチャートを用いて説明する。

【 0 2 5 5 】

図 1 8 は、図 1 5 の論理回路最適化装置 8 のフローチャートである。
図 1 8 に示すように、このフローチャートは、図 7 のフローチャートのステップ S 1 の前にステップ S 0 を付加したものである。

【 0 2 5 6 】

従って、ステップ S 0 にて、置換手段 9 5 は、メモリ置換テーブル情報を参照して、論理エミュレーション対象のメモリデバイスを実現するメモリデバイスを、回路長で表すことにより、論理エミュレーション対象のメモリデバイスを論理回路に置き換える。

【 0 2 5 7 】

こうすることで、論理エミュレーション対象のメモリデバイスを組み合わせ回路とみなすことができる。

【 0 2 5 8 】

その後は、論理回路最適化装置 8 は、実施の形態 1 と同様のステップ S 1 からステップ S 8 を行う。

【 0 2 5 9 】

なお、図 1 8 のステップ S 8 の代わりに、図 9 のステップ S 8 1 からステップ S 9 0 の処理を実行することもできる。

【 0 2 6 0 】

さて、以上のように、本実施の形態では、実施の形態 1 または実施の形態 2 と同様の構成を具備しているため、これらと同様の効果を奏する。

【0261】

さらに、本実施の形態では、論理エミュレーション対象のメモリデバイスを実現するメモリデバイスを回路長で表すことにより、論理エミュレーション対象のメモリデバイスを論理回路に置き換えている。

【0262】

このため、論理エミュレーション対象のメモリデバイスについても、通常の回路と同様に、回路分割手段85により、分割フリップフロップを挿入して、細分化できる。よって、より一層のFPGAの使用効率の向上を図ることができる。

【0263】

【発明の効果】

請求項1又は12記載の発明では、クラスタ長の長いクラスタにフリップフロップが挿入されるため、クラスタ長の長いクラスタが、クラスタ長の短い複数のクラスタに分割される。

【0264】

このため、論理エミュレーション装置の可変論理素子（例えば、FPGA）にクラスタを割り付ける際の自由度が高くなる。

【0265】

また、レンツ・ルールが示すような可変論理素子のI/Oピン数によるゲート搭載の非効率化が発生しないため、可変論理素子の使用効率を向上できる。

【0266】

以上により、論理エミュレーション装置に割り付ける論理回路の最適化が図られる。

【0267】

請求項2又は13記載の発明では、予め定められたクラスタ長を超えるクラスタについては、漏れなくフリップフロップが挿入されるため、最適化をより一層図ることができる。

【0268】

請求項3又は14記載の発明では、論理エミュレーション装置に割り付ける論理回路のより一層の最適化を図ることができる。

【0269】

請求項4又は15記載の発明では、論理エミュレーション対象のメモリデバイスについても、通常の回路と同様に、最適化処理を行うことができる。

【0270】

請求項5又は16記載の発明では、論理エミュレーション対象のメモリデバイスを実現するメモリデバイスが回路段数で表されるため、論理エミュレーション対象のメモリデバイスについても、通常の回路と同様に、最適化処理を行うことができる。

【0271】

請求項6又は17記載の発明では、論理エミュレーション対象のメモリデバイスを実現するメモリデバイスが信号の伝搬時間で表されるため、論理エミュレーション対象のメモリデバイスについても、通常の回路と同様に、最適化処理を行うことができる。

【0272】

請求項7又は18記載の発明では、回路段数の多いクラスタにフリップフロップが挿入されるため、回路段数の多いクラスタが、回路段数の少ない複数のクラスタに分割される。

【0273】

このため、可変論理素子にクラスタを割り付ける際の自由度が高くなり、可変論理素子の仕様効率の向上を図ることができる。

【0274】

請求項8又は19記載の発明では、信号の伝搬時間が長いクラスタにフリップフロップが挿入されるため、信号の伝搬時間が長いクラスタが、信号の伝搬時間が短い複数のクラスタに分割される。

【0275】

このため、可変論理素子にクラスタを割り付ける際の自由度が高くなり、可変論理素子の仕様効率の向上を図ることができる。

【0276】

請求項9又は20記載の発明では、可変論理素子（例えば、FPGA）に収ま

らないクラスタにフリップフロップが挿入されるため、可変論理素子に収まらないクラスタが、クラスタ長の短い複数のクラスタに分割される。

【0277】

このため、クラスタ長の短いクラスタを、可変論理素子に割り付けることができ、可変論理素子の使用効率を向上できる。

【0278】

以上により、論理エミュレーション装置に割り付ける論理回路の最適化が図られる。

【0279】

請求項10又は21記載の発明では、フリップフロップを挿入後でも、挿入前と同様の動作速度を維持できる。

【0280】

請求項11又は22記載の発明では、論理合成後のクラスタのクラスタ長が短くなるため、論理エミュレーション装置の可変論理素子（例えば、FPGA）にクラスタを割り付ける際の自由度が高くなる。

【0281】

また、レンツ・ルールが示すような可変論理素子のI/Oピン数によるゲート搭載の非効率化が発生しないため、可変論理素子の使用効率を向上できる。

【0282】

以上により、論理エミュレーション装置に割り付ける論理回路の最適化が図られる。

【0283】

さらに、ハードウェア記述言語段階でフリップフロップを挿入するため、論理合成後にフリップフロップを挿入する場合より、高速に処理することができる。

【図面の簡単な説明】

【図1】

本発明の実施の形態1における論理エミュレーションシステムのブロック図

【図2】

同回路割付装置のブロック図

【図 3】

同論理回路最適化装置のブロック図

【図 4】

(a) 同回路分割前のクラスタの例示図

(b) 同回路分割後のクラスタの例示図

【図 5】

同分割フリップフロップの挿入方法を説明するためのクラスタの例示図

【図 6】

同分割フリップフロップの挿入方法を説明するためのクラスタのタイミング図

【図 7】

同論理回路最適化装置のフローチャート

【図 8】

(a) 本発明の実施の形態 2 における論理回路最適化装置の説明図

(b) 同論理回路最適化装置の説明図

【図 9】

同論理回路最適化装置による最適化論理回路情報生成処理のフローチャート

【図 10】

本発明の実施の形態 3 における論理合成装置のブロック図

【図 11】

同機能分割フリップフロップ挿入前の HDL におけるモジュールの例示図

【図 12】

同機能分割フリップフロップ挿入前のモジュールのツリー図

【図 13】

同機能分割フリップフロップ挿入後の HDL におけるモジュールの例示図

【図 14】

同論理合成装置のフローチャート

【図 15】

本発明の実施の形態 4 における論理回路最適化装置のブロック図

【図 16】

同論理回路最適化装置の動作の説明図

【図17】

同メモリ置換テーブル情報の例示図

【図18】

同論理回路最適化装置のフローチャート

【図19】

従来の論理エミュレーション装置のブロック図

【図20】

(a) 従来のクラスタリング前の論理回路の概念図

(b) 従来のクラスタリング後の論理回路の概念図

【図21】

従来の論理エミュレーション装置のFPGAに割り付けられたクラスタの例示図

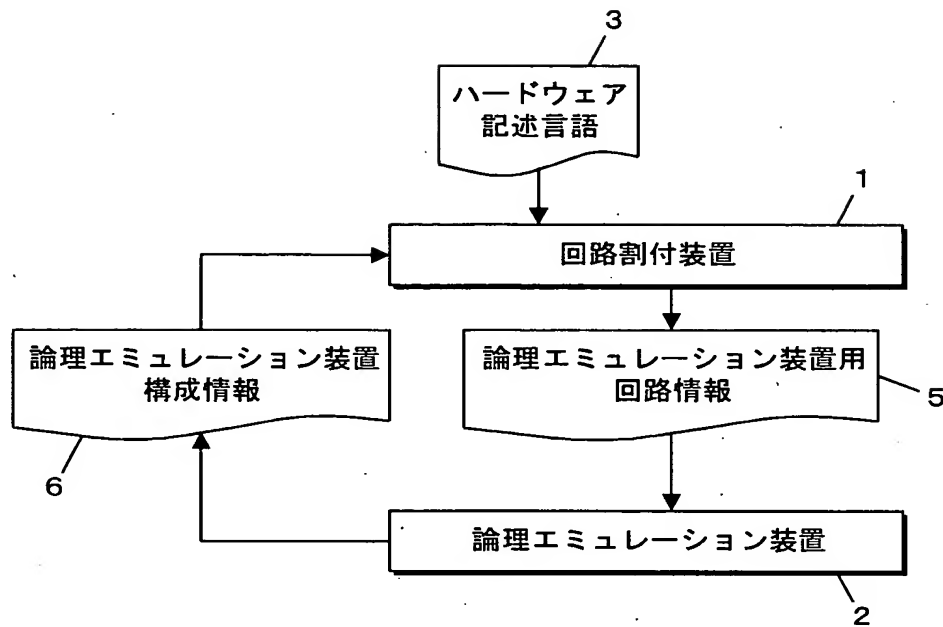
【符号の説明】

- 1 回路割付装置
- 2, 100 論理エミュレーション装置
- 3 ハードウェア記述言語
- 4 論理回路情報
- 5 論理エミュレーション装置用回路情報
- 6 論理エミュレーション装置構成情報
- 8 論理回路最適化装置
- 9 論理合成装置
- 10 FPGA割付装置
- 11 最適化論理回路情報
- 80 クラスタリング手段
- 81 クラスタリング論理回路情報記憶手段
- 82 クラスタ長測定手段
- 83 クラスタ長記憶手段
- 84 クラスタ選択手段

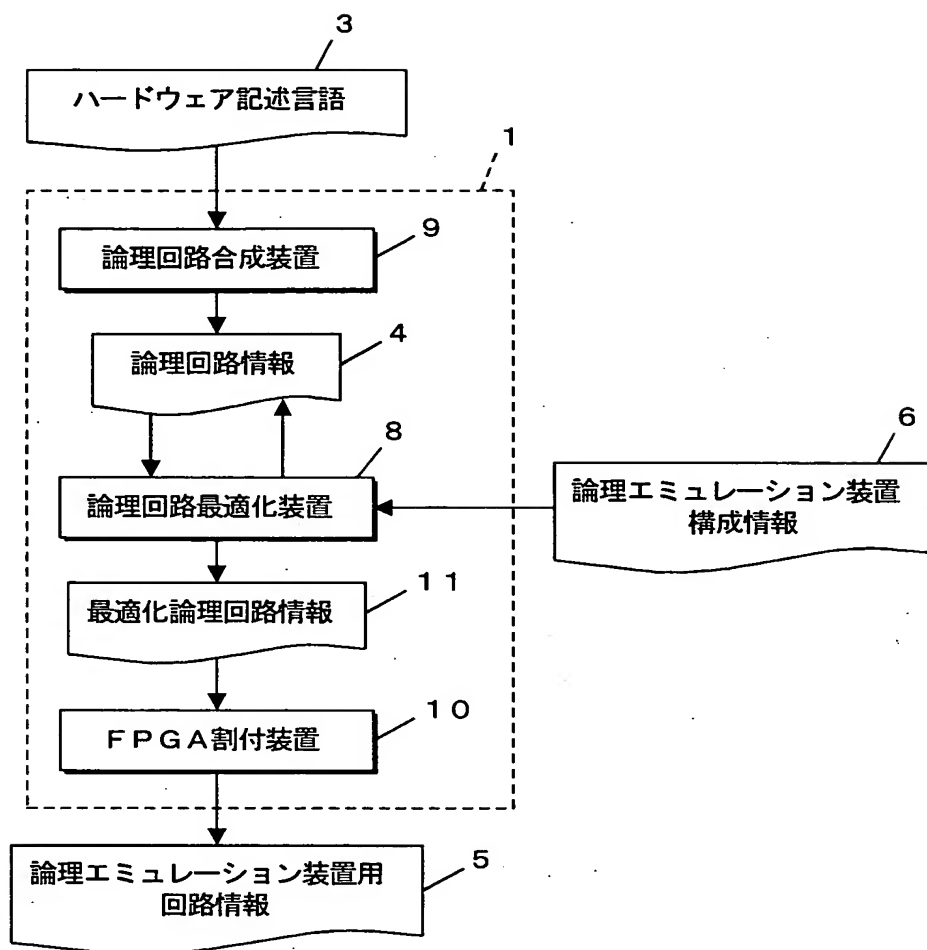
- 8 5 回路分割手段
- 8 6 回路分割論理回路情報記憶手段
- 8 7 クラスタ長判定手段
- 8 8 最適化論理回路情報生成手段
- 8 9 最適化論理回路情報記憶手段
- 9 0 モジュール分割手段
- 9 1 H D L 記憶手段
- 9 2 合成手段
- 9 3 演算素子数算出手段
- 9 5 置換手段
- 1 0 1 ~ 1 0 3 ロジックチップ
- 1 0 4 メモリモジュール
- 1 0 5 ユーザ指定チップ
- 1 0 6 相互接続チップ

【書類名】 図面

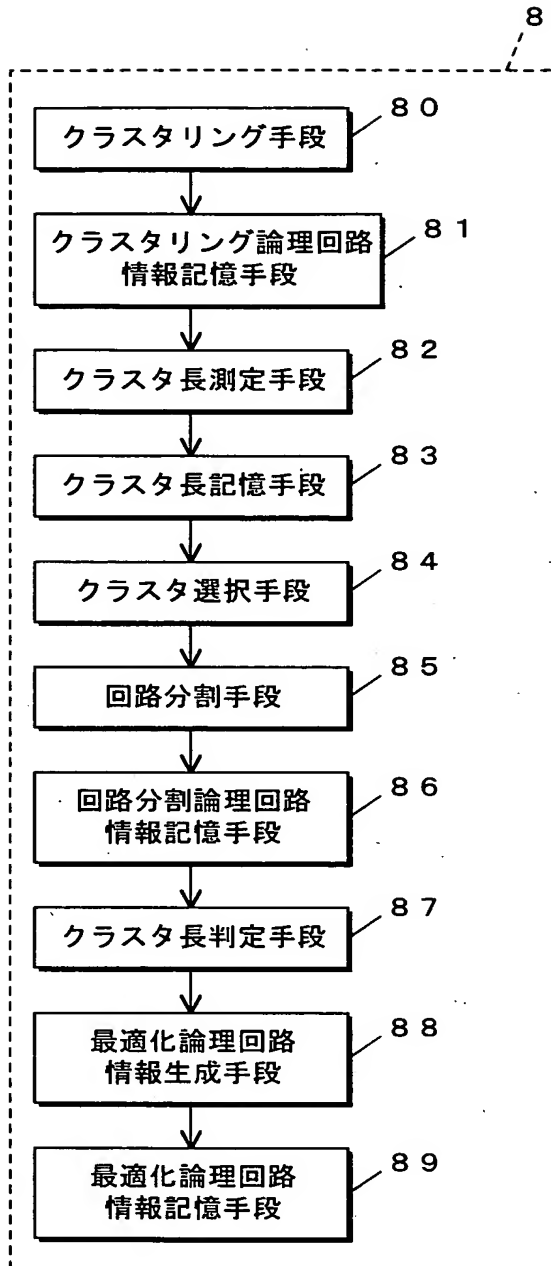
【図 1】



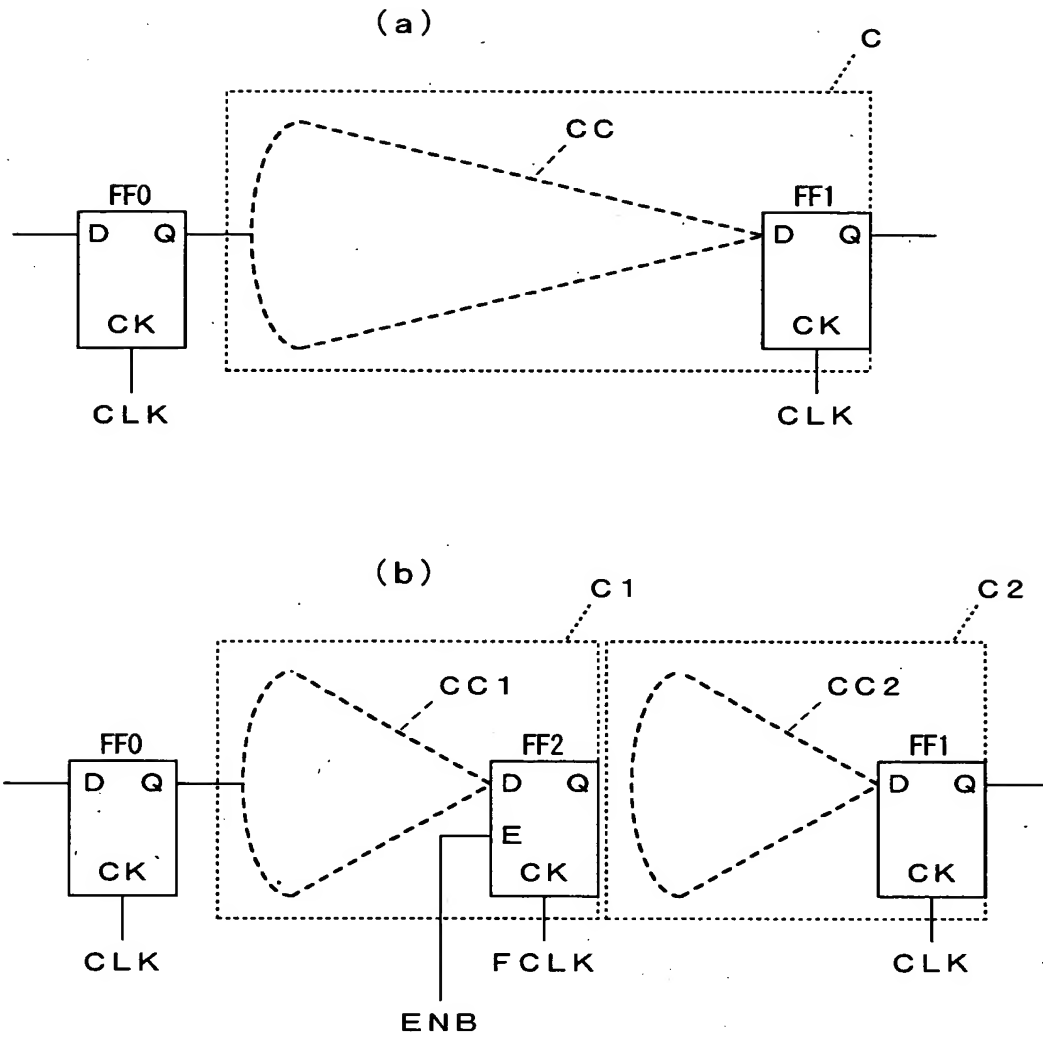
【図 2】



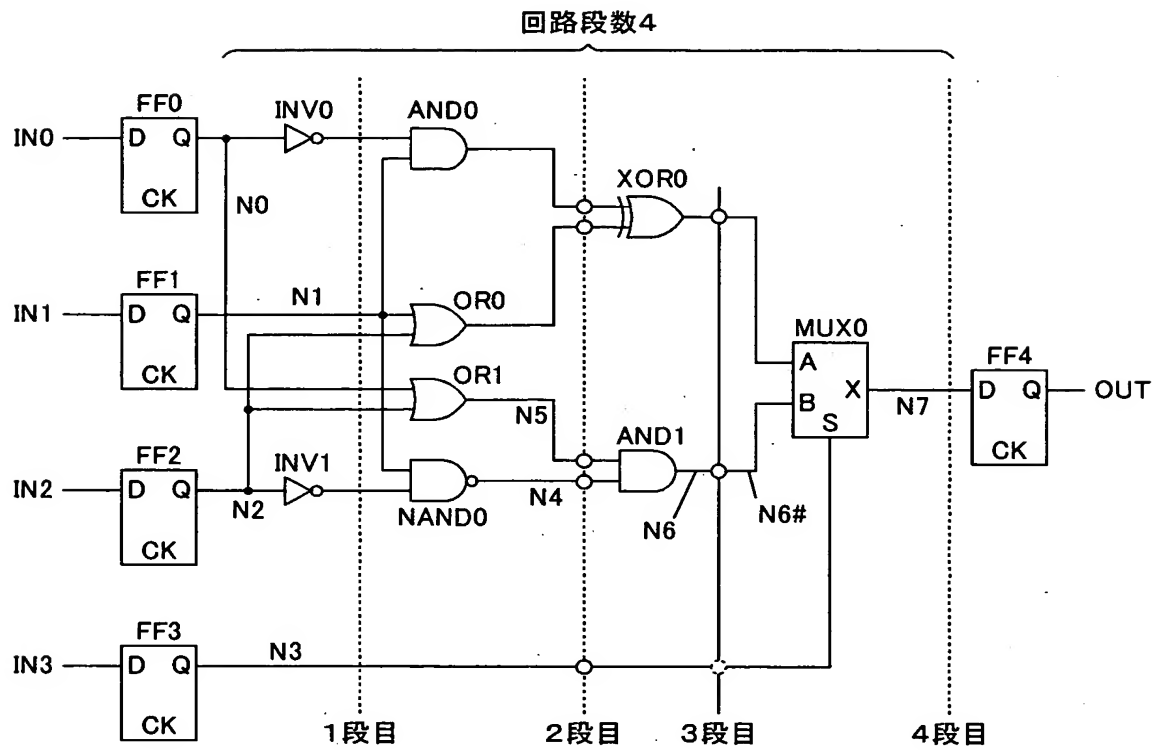
【図 3】



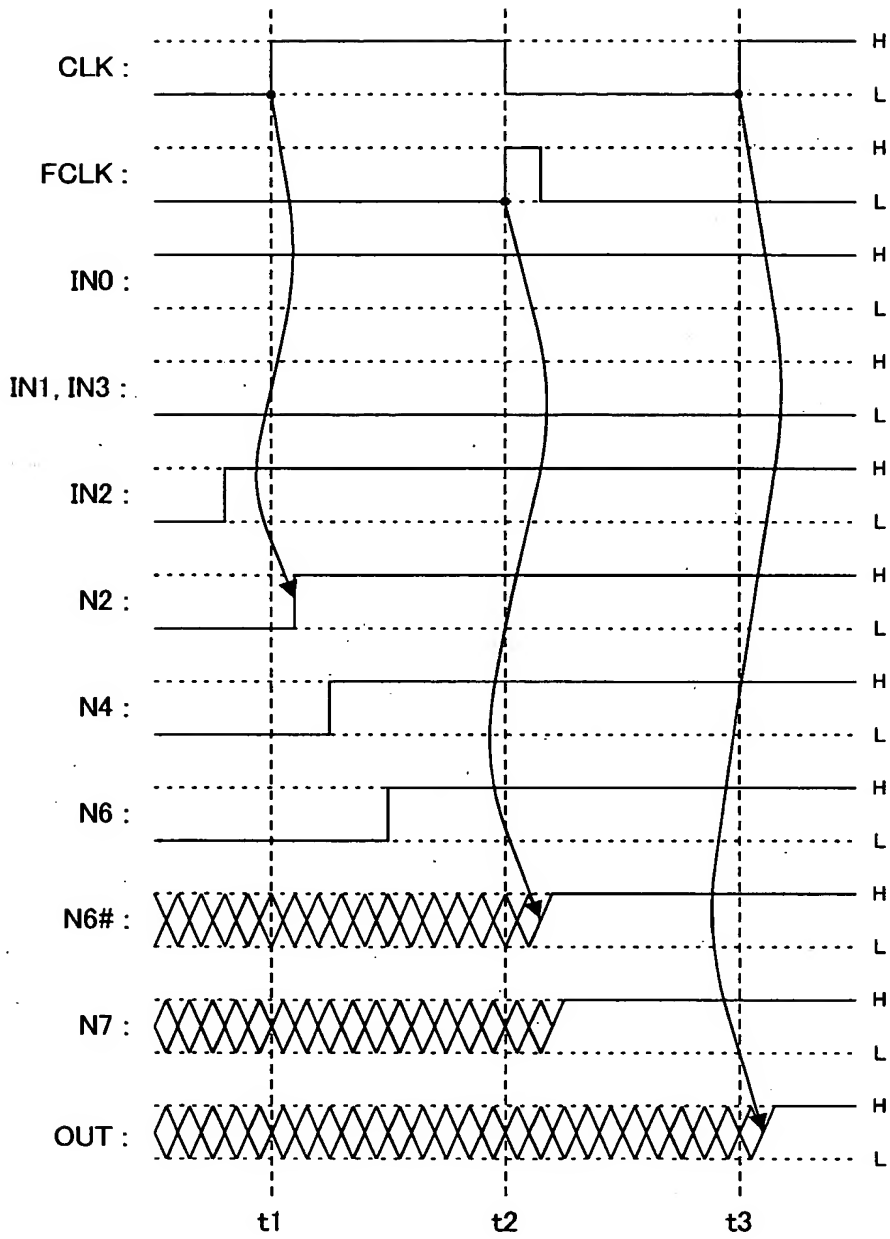
【図 4】



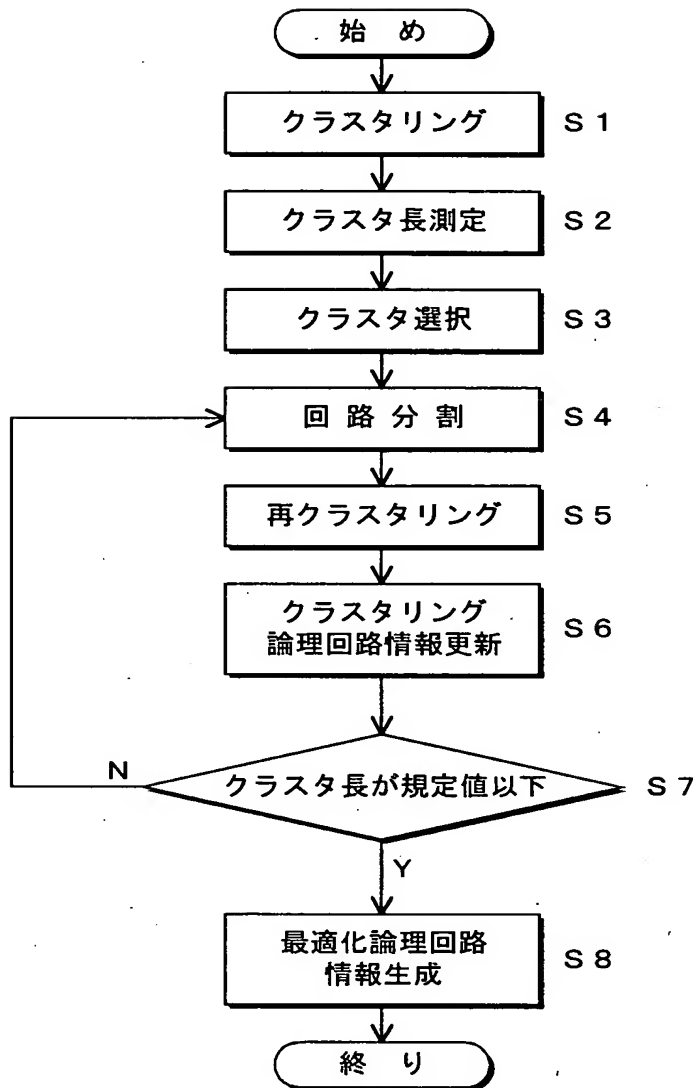
【図 5】



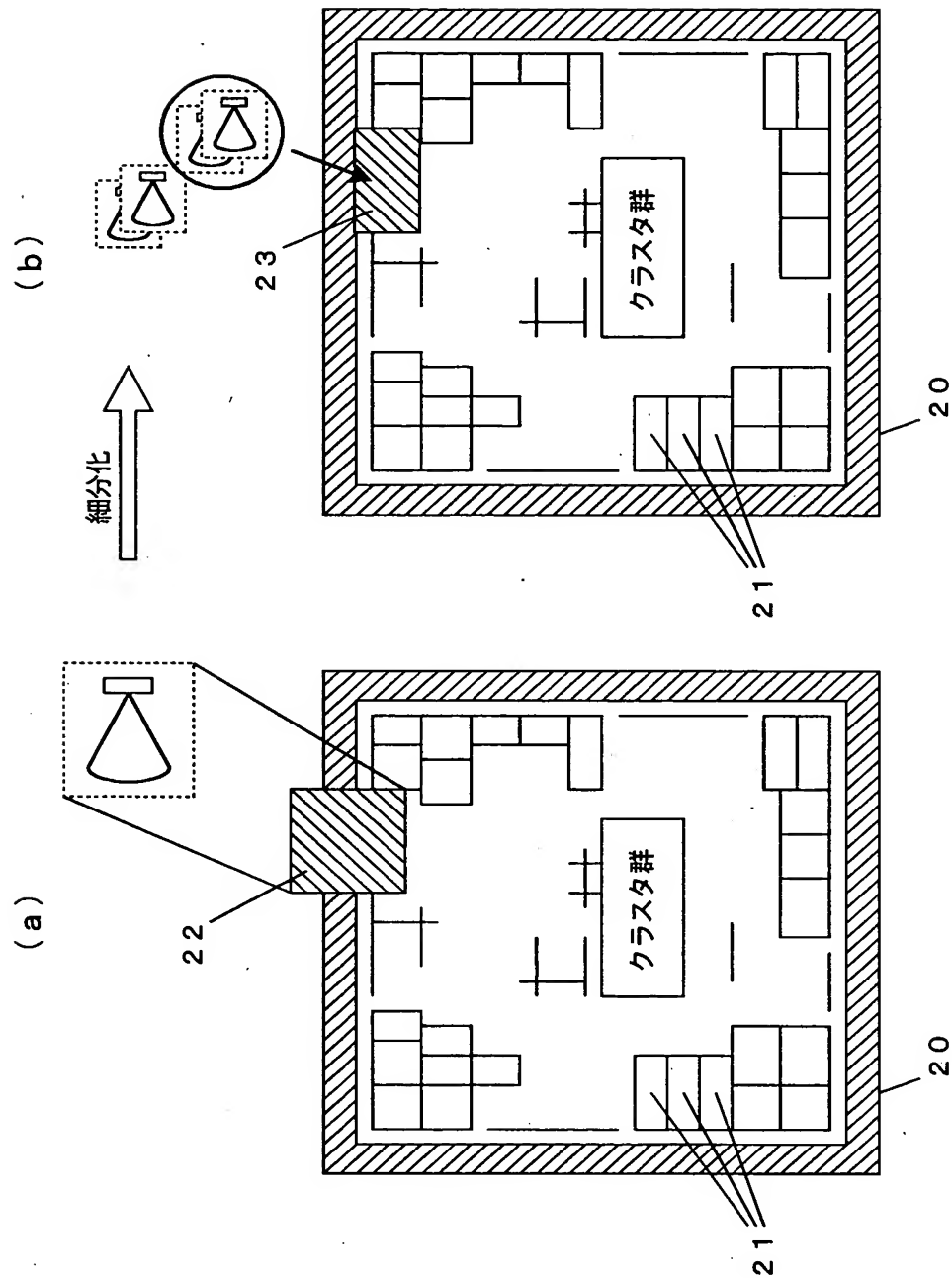
【図 6】



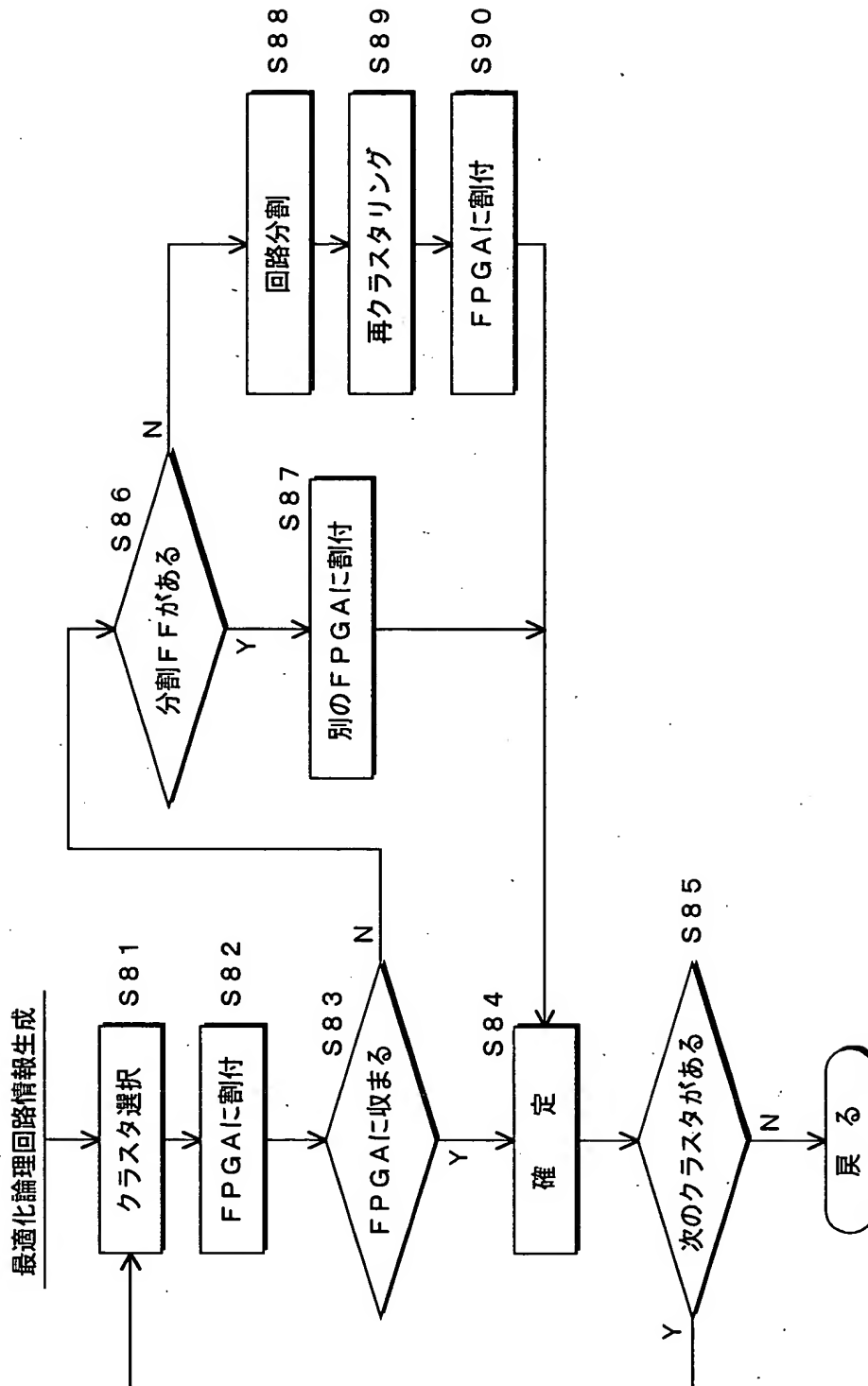
【図7】



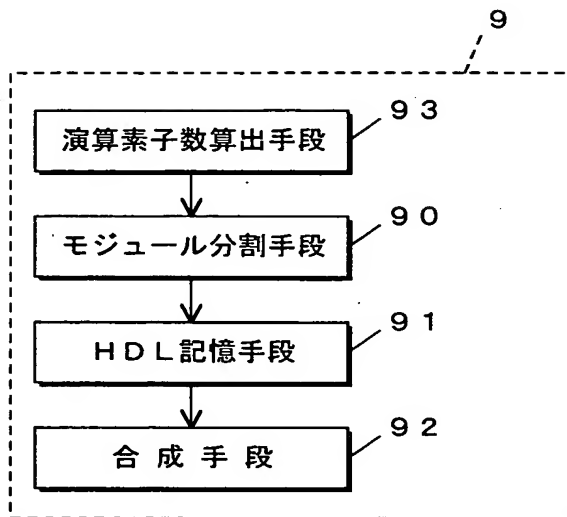
【図 8】



【図 9】



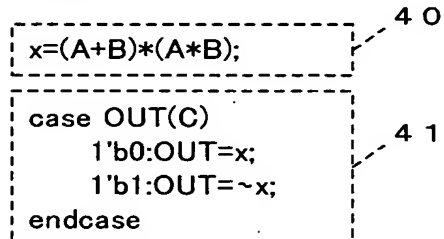
【図 1 0】



【図 1 1】

```

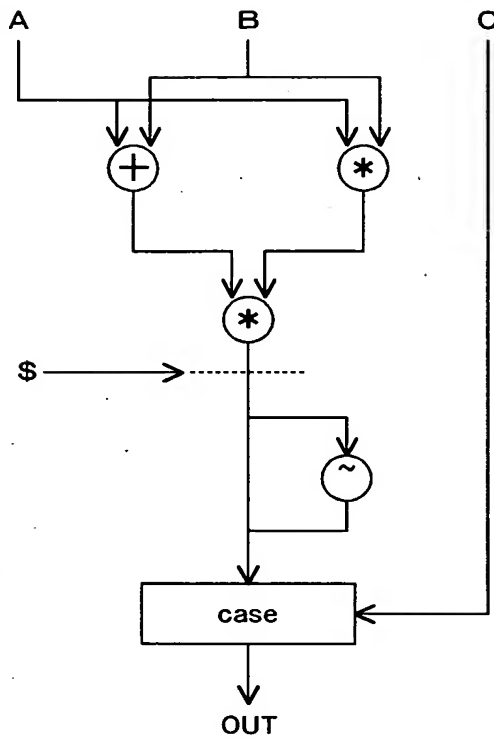
module example_module(A,B,C,OUT)
input A,B,C;
output OUT;
    
```



```

endmodule
    
```

【図 1 2】



【図 1 3】

```
module example_module(A,B,C,OUT)
input A,B,C;
output OUT;
```

```

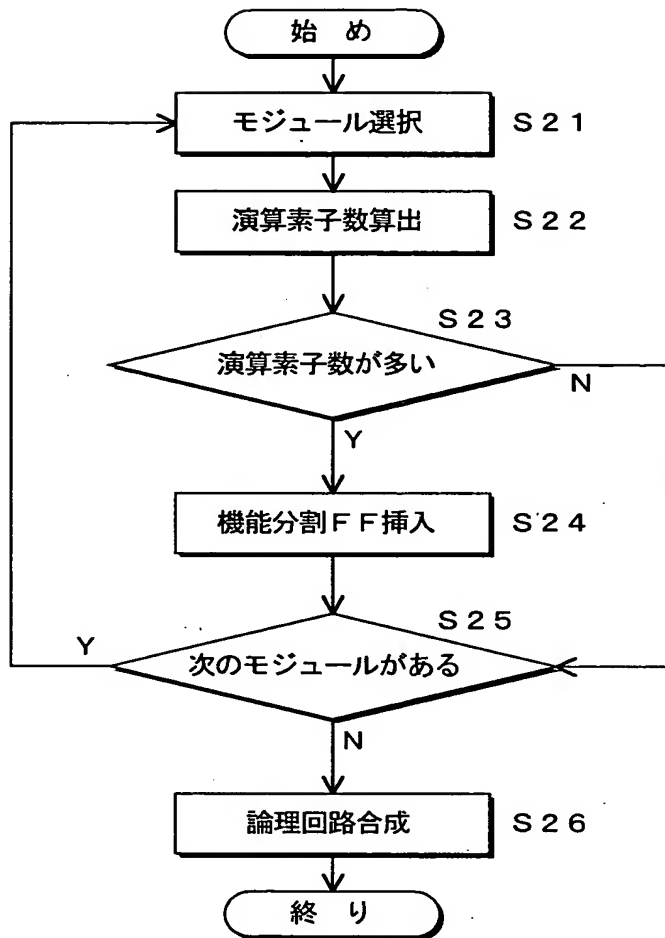
x=(A+B)*(A*B);
always(negedge CLK)
_x=x;
case OUT(C)
1'b0:OUT=_x;
1'b1:OUT=~_x;
endcase
endmodule
```

4 0

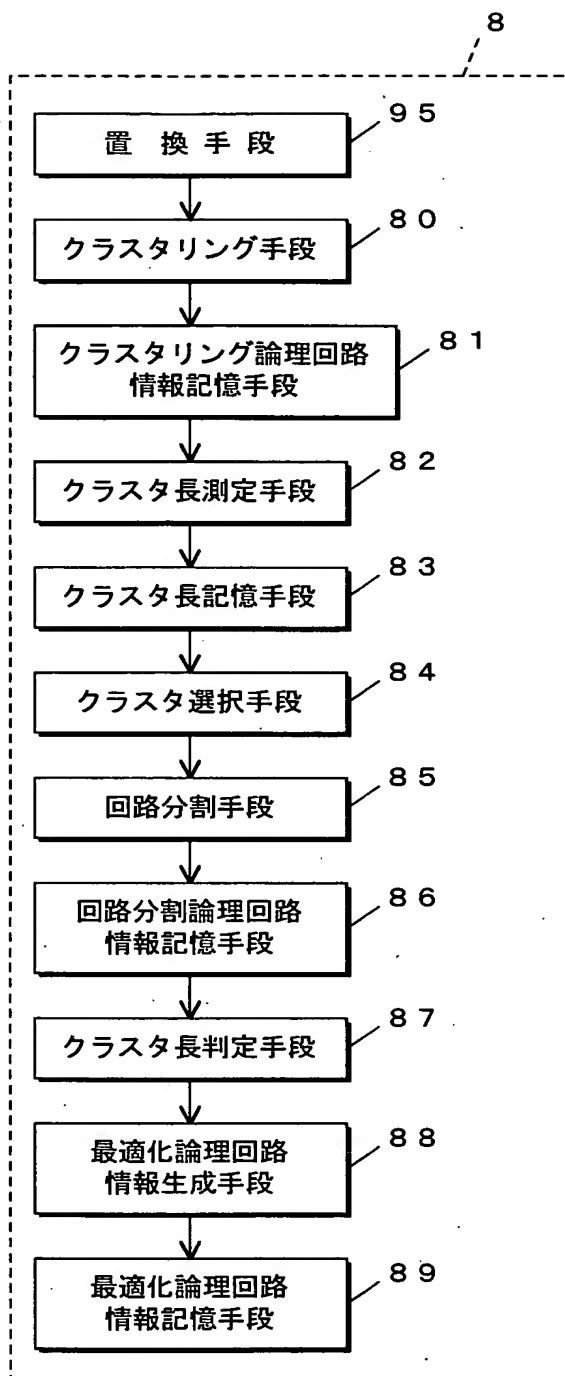
4 2

4 1

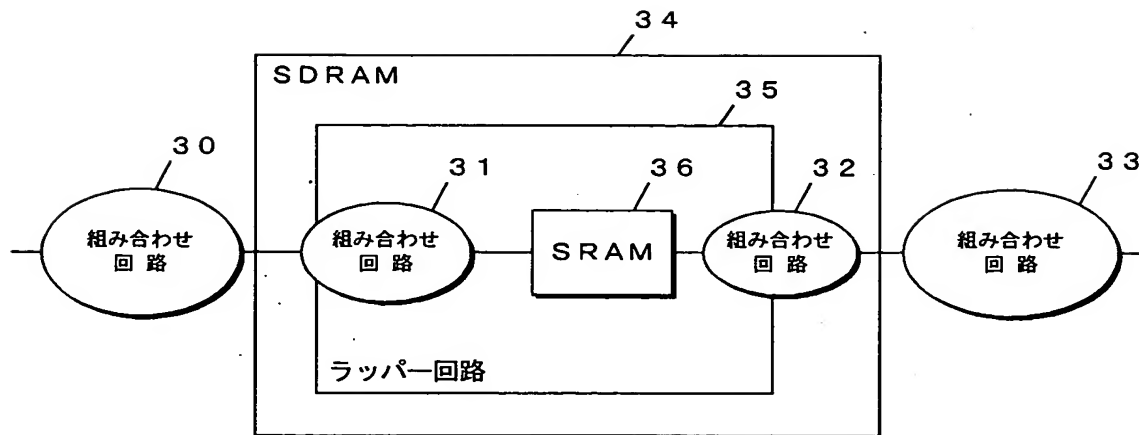
【図 1 4】



【図15】



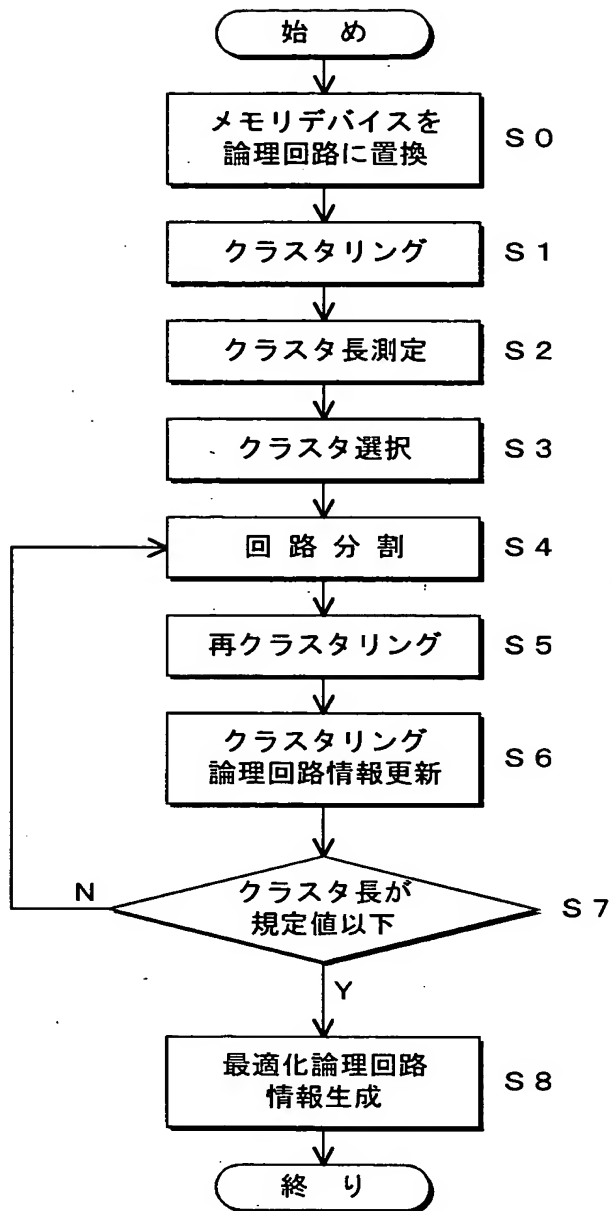
【図 16】



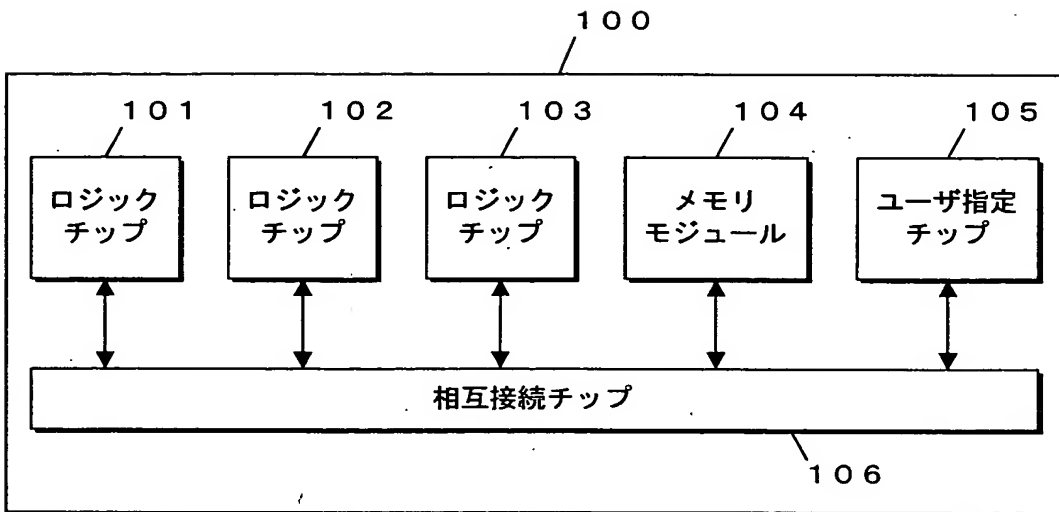
【図 17】

メモリ種類	換算値(回路段数)
4Mbit SRAM	10
1Mbit SRAM	2

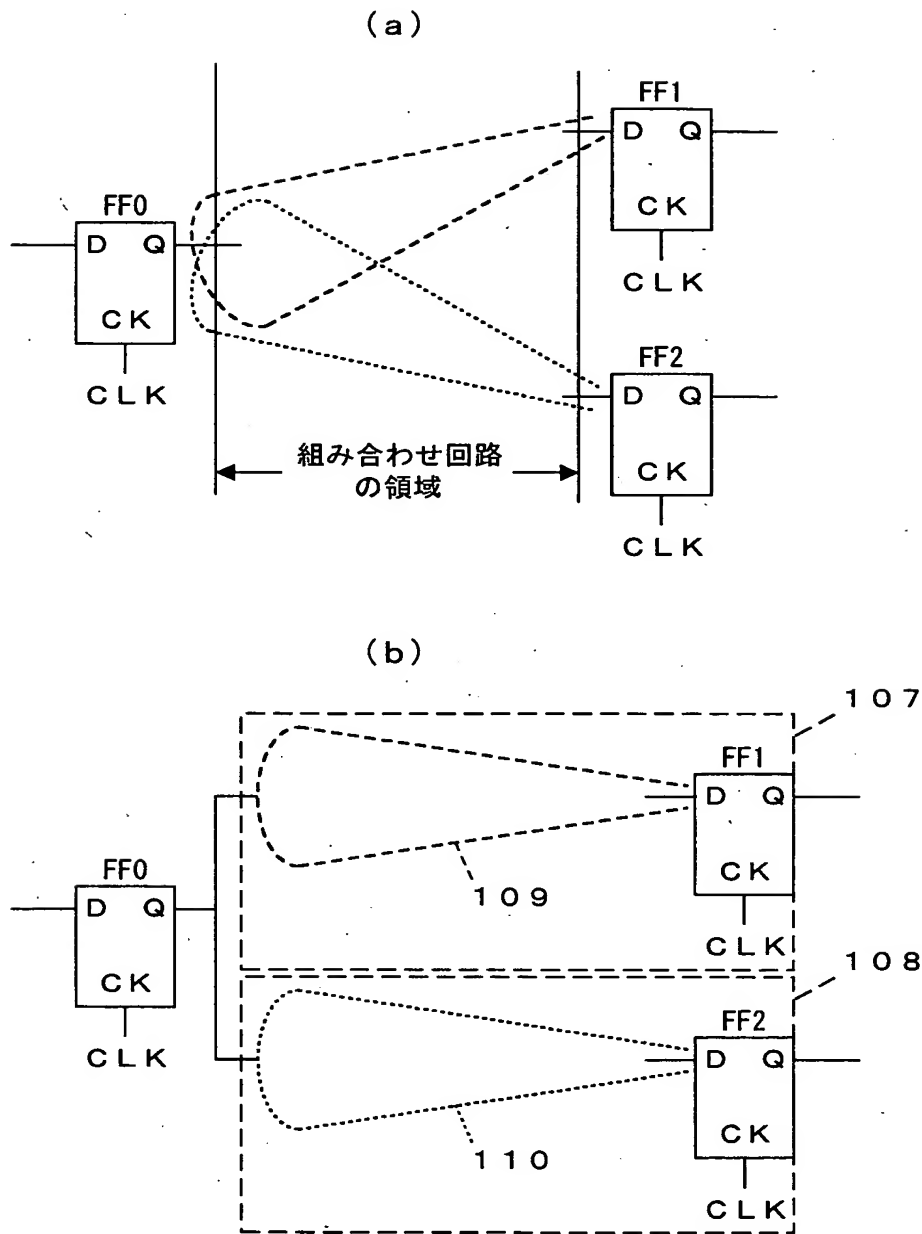
【図 18】



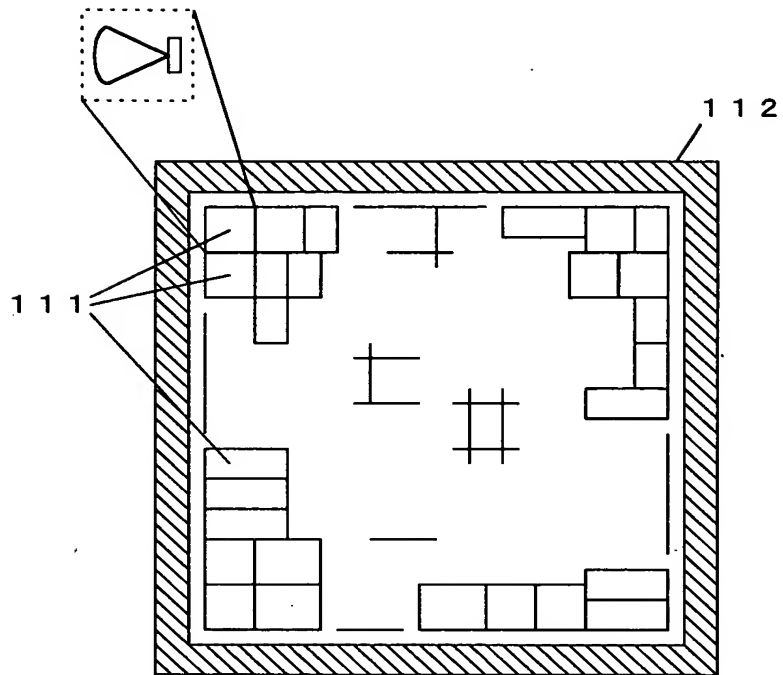
【図19】



【図 20】



【図 21】



【書類名】 要約書

【要約】

【課題】 論理エミュレーション装置の可変論理素子の使用効率を向上できる論理回路最適化方法を提供する。

【解決手段】 クラスタ長が予め定められたクラスタ長を超えるクラスタCに、分割フリップフロップFF2を挿入する。フリップフロップを挿入したクラスタCに対して、再クラスタリングを行い、細分化したクラスタC1, C2を得る。論理エミュレーション装置の可変論理素子（例えば、FPGA）にクラスタを割り付ける際の自由度を向上できる。

【選択図】 図4

出 願 人 履 歴 情 報

識別番号 [000005821]

1. 変更年月日 1990年 8月28日

[変更理由] 新規登録

住 所 大阪府門真市大字門真1006番地

氏 名 松下電器産業株式会社